UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIA DA COMPUTAÇÃO

SISTEMA DE DETECÇÃO DE INTRUSÃO BASEADO EM REDES NEURAIS

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação

Carlos Adriani Lara Schaeffer

Orientador: Prof. Dr. Mauro Roisenberg

Florianópolis, dezembro de 2003.

Folha de aprovação

AGRADECIMENTOS

À minha esposa Ana Paula e minha filha Ariell, aos meus pais e irmãos, ao meu orientador Prof. Dr. Mauro Roisenberg e aos colegas professores da Universidade de Passo Fundo. Pelo incentivo, motivação e apoio.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

CERT	"Computer Emergency Response Team"- Equipe especializada em incidentes computacionais
CIDF	"Common Intrusion Detection Framework"
CISL	"Common Intrusion Specification Language"
CPU	"Central Processing Unit" - Unidade Central de Processamento
DNS	"Domain Name System" – Sistema de Nomes de Domínio
GPL	"General Public License" – Licença Pública Geral
IDS	"Intrusion Detection System" – Sistema de Detecção de Intrusos
IETF	"Internet Engineering Task Force" – Força Tarefa de Engenheiros da Internet
IP	"Internet Protocol" – Protocolo para inter-rede
LAN	"Local Area Network" – Rede Local
MAN	"Metropolitan Area Network" – Rede metropolitana
RM-TCP/IP	"Reference Model TCP/IP" – Modelo de Referência TCP/IP
RNA	Rede Neural Artificial
SSH	"Security Socket Layer" – Camada de Conexão Segura
TCP	"Transmission Control Protocol" – Protocolo de Controle de Transmissão
UDP	"User Datagram Protocol" – Protocolo de Datagramas do Usuário
WAN	"Wide Area Network" – Redes de Longa Distância
WWW	"World Wide Web" – Larga Rede Mundial

RESUMO

Este trabalho apresenta um estudo de problemas relacionados com segurança de informações em redes de computadores. São apresentadas algumas técnicas utilizadas para tentar garantir a segurança das informações em um ambiente de redes de computadores, como ferramentas de criptografia, *Firewall* e Sistemas de Detecção de Intrusão(IDS), apresentados alguns conceitos importantes na área de segurança da informação, alguns ataques conhecidos e algumas medidas preventivas. São descritos e classificados vários modelos de Sistemas de Detecção de Intrusão em redes de computadores. É feito um estudo de Redes Neurais Artificiais que será utilizada para avaliação de padrões de comportamento e detecção de padrões intrusivos. É feita a escolha de um modelo para análise e testes em um laboratório real utilizados por um grupo de professores da Universidade de Passo Fundo. A partir do uso desta rede, são criados padrões de comportamento e coletados registros de comandos executados por estes usuários a fim de verificar a existência de padrões de comportamento suspeito com o apoio de uma rede neural artificial. Para finalizar, é apresentada uma avaliação da análise feita por esta rede neural, relatando a fase de treinamento e alimentação desta rede e os resultados obtidos.

ABSTRACT

This work presents a study of problems related with information security on networks. Some techniques are presented used to try to guarantee the security security in an environment of computer networks, as cryptography tools, Firewall and Intrusion Detection System(IDS), and some important concepts in the area of information security, some known attacks and some measured preventive. Several models of Intrusion Detection Systems in networks are described and classified. It is made a study of Artificial Neural Networks to be used to evaluate the behavior patterns and detection of intrusive patterns. It is made the choice of a model for analysis and tests in a real laboratory used by the users group of University of Passo Fundo. Starting from the use of this network, the behavior patterns are created and collected registers of commands executed for these users in order to verify the existence of suspected behavior patterns with the support of an artificial neural network. To conclude, an evaluation of the analysis made for this neural network is presented, reporting the results of this phase.

1. INTRODUÇÃO

1.1. Motivação

Verificando a história da computação, nota-se um aumento significativo de microcomputadores na vida de empresas e pessoas. Nas grandes empresas, onde haviam sistemas centralizados em grandes computadores chamados *mainframes*, aconteceu um processo de descentralização, aumentando consideravelmente o uso da microinformática. Em médias empresas onde se usavam poucos computadores, hoje em dia, existem vários equipamentos ligados em rede para as mais diversas tarefas. Nas pequenas empresas, que há poucos anos atrás não possuíam sistemas informatizados, nota-se que hoje, a grande maioria adquiriu computadores e sistemas. Até mesmo para uso pessoal, o microcomputador se tornou bastante popular nas mais diversas atividades, inclusive para diversão e estudo de crianças e adultos. Isto se deve ao fato da queda nos custos destes equipamentos, na constante e rápida evolução técnica e também aos novos sistemas operacionais com interface gráfica amigável que simplificam o uso destes equipamentos para pessoas com pouco treinamento.

Mas sem dúvida, o grande avanço foi dado com a popularização da Internet, que permite a interligação de computadores em diversos ambientes, sejam profissionais ou pessoais, nos ambientes acadêmicos ou para diversão, para comércio ou para indústria e principalmente para disseminar informações. O mundo inteiro passou a dispor de recursos computacionais e serviços das mais diversas formas e nos mais diversos meios.

A facilidade de acesso a esta rede mundial trouxe muitos benefícios, mas também muitos problemas. Entre estes, alguns problemas sociais como a exclusão digital e a questão dos direitos autorais, mas um dos problemas mais graves e que afetam todas as áreas, citamos o problema da segurança das informações que trafegam nesta rede ou que estão armazenadas nos seus computadores servidores.

Inúmeras atividades são efetuadas utilizando os recursos computacionais das redes de computadores, necessitando serviços confiáveis e em muitas vezes troca de informações sigilosas. O funcionamento correto destes recursos torna-se de fundamental importância.

Paralelamente, ações de pirataria, tentativas de intrusão e invasões consumadas aumentam a cada dia, envolvendo um número cada vez maior de máquinas. Com isto, torna-se necessário e indispensável, o uso de técnicas especiais de segurança. O risco tende a aumentar, pois, com a disseminação de informações, houve um significativo avanço no entendimento de como operam os sistemas. Isso fez com que intrusos se tornassem verdadeiros especialistas em determinar e explorar vulnerabilidades a fim de obterem acessos privilegiados. Estes mesmos intrusos também passaram a criar e utilizar técnicas de intrusão de difícil rastreamento e identificação.

Atualmente, são utilizadas diversas técnicas de segurança, como as barreiras que protegem o perímetro da rede, conhecidas como *firewall,* (parede corta-fogo) uma das técnicas mais utilizadas. Porém, se um atacante conseguir ultrapassar esta barreira, ou ainda, o atacante já estiver dentro da rede, como por exemplo, um funcionário descontente ou mal intencionado, de nada adiantará a utilização de *firewalls.* Com isto, surgiram os primeiros estudos de técnicas voltadas à proteção interna da rede, os primeiros Sistemas de Detecção de Intrusão. Os IDS, com são chamados, verificam o comportamento interno da rede, como análise de tráfego feita pelos IDS baseados em rede, e análise do comportamento dos usuários e sistemas, executada pelos IDS baseados em máquina. Estas análises costumam ser feitas tentando detectar atacantes por abuso ou por anomalia, tentando comparar o padrão de comportamento com os padrões de ataques conhecidos.

A maioria dos IDS atuais trabalham comparando os padrões de ataques conhecidos, também chamados de assinaturas de ataques, com os dados recolhidos de registros de auditoria ou das informações contidas nos protocolos que estão trafegando na rede. Nesta situação reside um grande problema, pois quando surge um novo tipo de ataque, ou mesmo uma variação de um ataque já conhecido, o IDS acaba não encontrando um padrão de assinatura de ataque e deixa de detectar este ataque. Com isso, os administradores da rede, ficam bastante inseguros quanto a confiabilidade e na

dependência de atualizações dos IDS pelos seus fabricantes assim que novos ataques ou vulnerabilidades forem sendo descobertos.

1.2. Objetivo

Este trabalho apresenta um sistema de detecção de intrusos baseado em comportamento utilizando uma rede neural como método de auxílio na avaliação de registros, incrementando a análise de comportamento generalizando ao máximo as assinaturas de ataque permitindo assim, a detecção de comportamento anômalo.

O objetivo deste trabalho é desenvolver os métodos de análise dos sistemas de detecção de intrusão com a utilização de redes neurais, para isso, foi utilizado o laboratório do curso de Ciência da Computação da Universidade de Passo Fundo onde foi realizado um monitoramento do comportamento de alguns usuários, registrando os comandos utilizados por estes em suas sessões de trabalho traçando um perfil de comportamento.

1.3. Estrutura do Documento

Esta dissertação apresenta no capítulo 2 uma introdução aos conceitos básicos de redes de computadores e do modelo de referência TCP/IP, no capítulo 3 são apresentados os principais conceitos e problemas da área de segurança de redes de computadores e suas informações e recursos, bem como algumas técnicas de proteção mais utilizadas. No capítulo 4, são abordados os conceitos de Sistemas de Detecção de Intrusão, sua classificação, características dos IDS, e uma descrição de um IDS, baseado em licença GPL, mais utilizado em redes de médio porte baseada em Linux, o Snort. No capítulo 5, os conceitos básicos de redes neurais e alguns trabalhos existentes envolvendo Sistemas de Detecção de Intrusão com Redes Neurais. O capítulo 6 apresenta a proposta desta dissertação e os resultados obtidos na avaliação de comportamento anormal de usuários em um ambiente monitorado por um IDS baseado em servidor. Também são apresentados os direcionamentos futuros desta pesquisa para a utilização de redes neurais em IDS baseado em rede, analisando o tráfego de rede e o capítulo 7 conclui o trabalho.

2. REDES DE COMPUTADORES

A história nos mostra uma evolução impressionante no desenvolvimento computacional. Há poucos anos, duas ou três décadas, considerava-se ficção científica a idéia de milhões de computadores, com processadores menores que um selo postal, mas com alto poder de processamento, estarem distribuídos pelo mundo todo, em pequenas, médias e grandes empresas, em escritórios, em casas e até mesmo em equipamentos portáteis.

Nos anos 50, computadores eram máquinas complexas operadas por pessoal altamente especializado e que ocupavam salas inteiras. Os usuários destes computadores submetiam seus programas para processamento através de tarefas sem interação com o processamento.

Em meados dos anos 60, começaram a surgir as primeiras tentativas de interação com o computador. Utilizando uma técnica chamada t*ime-sharing*, onde usuários compartilhavam o processamento utilizando terminais conectados a um computador central através de uma linha de comunicação. Este foi o ponto de partida para o estabelecimento de necessidades de comunicação entre computadores. O crescente aumento do número de usuários compartilhando um único computador para a resolução de uma grande diversidade de problemas implicava numa necessidade crescente de atualizações e incremento na capacidade de cálculo e de armazenamento nas CPUs, o que nem sempre era viável ou possível, dado que os computadores do tipo *mainframes* nem sempre eram adaptados para suportar determinadas extensões.

Até o final dos anos 70, a informática apresentava-se através de grandes computadores centralizados em salas climatizadas. Era atual o termo Centro de Processamento de Dados ou CPD, que apenas grandes empresas, órgãos governamentais e algumas importantes universidades possuíam. Nesta época, começaram a surgir os

primeiros microcomputadores que revolucionariam o mercado com preços cada vez mais baixos e tecnologia cada vez mais sofisticada e miniaturizada. Começava então a descentralização do processamento, agora distribuído em diversas CPUs espalhadas pelos departamentos, originando assim um problema grave na hora de transferir e centralizar as informações. Surgia então, a necessidade de comunicação entre os computadores para que pudessem transferir arquivos e compartilhar recursos, como discos e impressoras.

Paralelamente, a tecnologia de comunicações alcançava a transmissão digital em linhas telefônicas através de *modems.* Este serviço era caro e apenas suportado por grandes companhias, uma vez que utilizavam linhas telefônicas de forma dedicada. Esta situação perdurou por algum tempo (no Brasil, até março de 1985) e era necessária outra solução para comunicação através de uma nova tecnologia de comunicação. Com a união entre a tecnologia de comunicação e a tecnologia de informação, surgiram as redes de computadores para mudar e revolucionar o mundo em que vivemos, criando novas formas de comunicação e incrementando o alcance dos sistemas computacionais.

Segundo Elizabeth Specialski (2002), o termo rede de computadores designa um conjunto de computadores autônomos e interconectados. Dois computadores são interconectados quando podem trocar informações através de algum mecanismo de comunicação. Quando se diz que eles devem ser autônomos deseja-se excluir os sistemas onde existe uma clara relação mestre/escravo. Hoje, em empresas que possuem computadores em diversos setores, apesar dos computadores serem autônomos e poderem trabalhar de forma independente, fica claro que a ligação entre os mesmos torna o desenvolvimento das tarefas mais eficiente.

A interligação entre computadores é feita para permitir que cada usuário tenha acesso a todos os recursos necessários para o desenvolvimento de sua tarefa sem importar-se com o tamanho de uma empresa ou mesmo com limites físicos. O objetivo destas conexões é o compartilhamento de recursos.

2.1. Tópicos importantes em redes de computadores

Algumas questões devem ser levadas em consideração quando se pensa em implementar uma rede de computadores. A primeira delas é a redução de custos, já que

os microcomputadores apresentam uma relação custo/benefício inferior aos equipamentos de grande porte. A escalabilidade, que é a possibilidade de incrementar gradualmente o desempenho e quantidade dos servidores, meios de comunicação, e estações de trabalho conforme o volume de trabalho da rede informações cresce. A confiabilidade, talvez o fator mais importante a ser considerado, significa que os dados e sistemas devem estar disponíveis e íntegros a qualquer momento em que se precisa deles.

2.2. Categorias de Redes

As redes de computadores permitem que sejam conectados computadores uns aos outros em um ambiente de trabalho ou à distância, desta forma, podemos classificar as redes de computadores pela sua escala e abrangência. Dividimos então, em três categorias: Redes locais, privadas, com abrangência de alguns metros até 1 Km, normalmente abrangendo os computadores de uma sala, um prédio ou campus; redes metropolitanas podendo ser privadas ou públicas, com distâncias metropolitanas entre os nós; e redes de longa distância, públicas ou privadas, interligando redes entre prédios, cidades ou até mesmo, países.

2.3. Arquitetura de Redes

As redes de computadores podem ser caracterizadas por diferentes topologias e configurações, com o objetivo único de transferir dados. Para viabilizar o funcionamento das redes, devem ser implementados mecanismos e procedimentos corretamente especificados. Estes mecanismos são implementados para resolver diversas funções em uma rede de computadores.

Os sistemas de rede foram então projetados, de forma estruturada, em níveis posicionados um acima do outro, onde cada nível desempenha tarefas específicas relacionadas ao projeto de rede. Em cada nível da estrutura, são implementados programas seguindo certas regras, chamados de protocolos de rede. Cada um destes objetos, trabalhando em uma camada, desempenha funções específicas daquele nível, comunicando-se com os protocolos de nível adjacente sem que a implementação de um nível seja conhecida por outro. Basicamente, um protocolo é um conjunto de regras sobre o modo como se dará à comunicação entre as partes envolvidas.

Este tipo de organização em camadas é conhecido como arquitetura da rede, e pode ser organizada em diversas camadas desde que contenham todas as especificações para permitir o correto desenvolvimento da rede. Devem conter especificações dos programas e dos equipamentos e materiais utilizados.

Uma mensagem que deve ser enviada para uma outra máquina da rede, irá ser tratada por todas as camadas do software de rede, conforme exemplo da Figura 1 começando pela camada de nível mais alto. A mensagem será tratada pela camada sete que passará para a camada seis. Todas as questões que dizem respeito ao projeto de uma rede, estarão implementadas nos protocolos de cada camada. Não existe uma comunicação física entre protocolos de mesmo nível em máquinas diferentes, apenas o nível mais baixo, o nível físico, é que faz a ligação física.

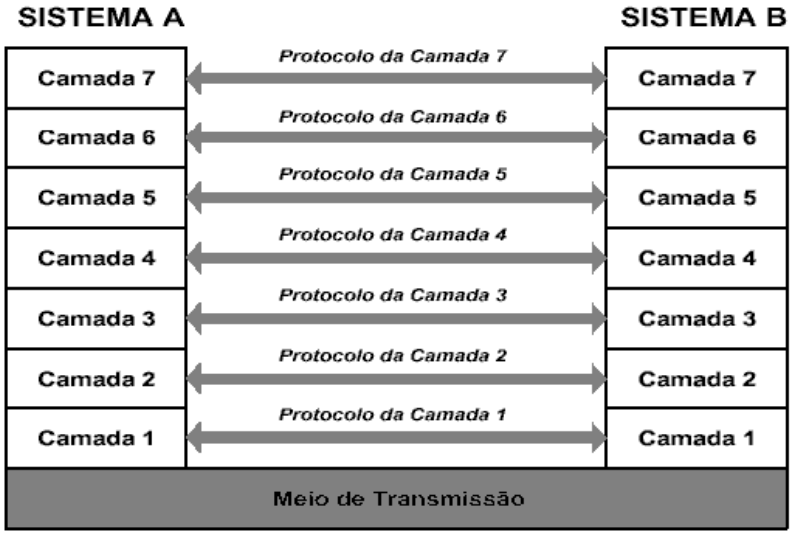

Figura 1 - Arquitetura de rede em camadas

2.4. Modelo de Referência OSI/ISO

A ISO (“International Organizational for Standardization”) criou, entre 1978 e 1984, um modelo de referência para arquitetura de redes em camadas, o RM-OSI (Reference Model – Open Systems Interconection) Modelo de Referência para Interconexão de Sistemas Abertos. Este modelo introduziu o conceito de Sistema Aberto, definido como “o sistema capaz de suportar os padrões de comunicação OSI de modo a interfuncionar com outros sistemas abertos de diferentes fornecedores”. Este modelo possui sete camadas: aplicação, apresentação, sessão, transporte, rede, enlace e física, cujos princípios de definição foram os seguintes:

?? Cada camada corresponde a um nível de abstração necessário ao modelo;

?? Cada camada possui suas funções próprias e bem definidas;

?? As funções de cada camada foram escolhidas segundo a definição dos protocolos normalizados internacionalmente;

?? A escolha das fronteiras entre cada camada deveriam ser definidas de modo a minimizar o fluxo de informação nas interfaces;

?? O número de camadas deve ser suficientemente grande para que funções distintas não precisem ser colocadas na mesma camada, e ser suficientemente pequeno para que a arquitetura não se torne difícil de controlar.

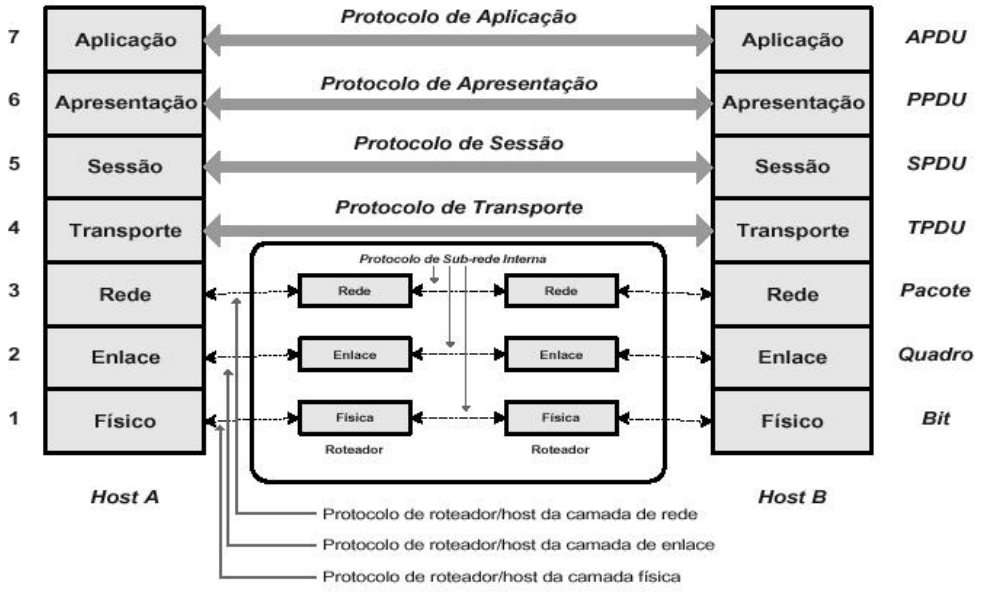

Figura 2 - Modelo de Referência OSI/ISO – Sete camadas

2.5. Modelo de Referência TCP/IP

A arquitetura TCP/IP surgiu com a criação de uma rede patrocinada pelo DoD (Department of Defense) - Departamento de Defesa dos Estados Unidos. Uma das tarefas essenciais dessa rede seria manter a comunicação entre alguns órgãos do governo e universidades, numa ocorrência de guerras ou catástrofes que afetassem os meios de comunicação daquele país. Dessa necessidade, surgiu a ARPANET, uma rede

que permaneceria intacta caso um dos servidores perdesse a conexão. A ARPANET necessitava então de um modelo de protocolos que assegurasse tal funcionalidade esperada, mostrando-se confiável, flexível e de fácil implementação. A partir daí foi então desenvolvida a arquitetura TCP/IP, que se tornou um padrão de fato. A ARPANET cresceu e tornou-se a rede mundial de computadores – Internet. A utilização (e facilidades) do padrão TCP/IP utilizado pelos fabricantes de outras redes, com a finalidade da conectividade com a Internet. A normalização do TCP/IP chegou após a sua utilização em massa.

A Internet cresceu, abrangendo centenas de redes individuais localizadas nos Estados Unidos e na Europa. Conectou aproximadamente 20.000 computadores de universidades, órgãos públicos e laboratórios de pesquisa organizacional. O tamanho e a utilização da Internet continuou em ascensão muito mais acelerada do que o previsto. No final de 1987, estimou-se que o crescimento alcançara 15% ao mês. Em torno de 1994, a Internet alcançava mais de 3 milhões de computadores em 61 países.

A utilização de protocolos TCP/IP e o crescimento da Internet não se limitaram a projetos financiados pelo governo. Grandes companhias voltadas para o setor de computadores conectaram-se à Internet, bem como muitas outras organizações de grande porte como companhias de petróleo, indústria automobilística, empresas de eletrônica, companhias farmacêuticas e portadoras de telecomunicações. As empresas de pequeno e médio porte começaram a conectar-se na década de 1990. Além disso, muitas outras utilizavam os protocolos TCP/IP em suas interligações de redes corporativas, mesmo tendo optado por não participar da Internet.

Uma expansão acelerada trouxe problemas de escala não previstos no projeto original e motivou os pesquisadores a encontrar técnicas para gerenciar numerosos recursos distribuídos. No projeto original, por exemplo, os nomes e endereços de todos os computadores conectados à Internet eram mantidos em um único arquivo que era editado manualmente e, a seguir, distribuído a todos os *sites* da Internet. Em meados da década de 1980, tornou-se óbvio que um banco de dados de origem não seria suficiente. Primeiro, os pedidos para atualização de arquivos rapidamente provocariam excesso dos recursos disponíveis para processá-los. Segundo, ainda que existisse um arquivo-fonte

correto, a capacidade da rede seria insuficiente para permitir a distribuição freqüente para todos os *sites* ou o acesso on-line a cada *site.*

Novos protocolos foram desenvolvidos e um sistema de atribuição de nome foi colocado em vigor através da Internet para permitir que qualquer usuário, automaticamente determinasse o nome de uma máquina remota. Conhecido como DNS (*Domain Name System)* Sistema de Nomes de Domínio, o mecanismo conta com máquinas denominadas "servidores de nome" para responder as consultas sobre nomes. Nenhuma das máquinas contém todo o banco de dados de nomes de domínios. Em vez de uma máquina, os dados são distribuídos por um conjunto de máquinas que utilizam protocolos TCP/IP para se comunicarem entre si quando estiverem respondendo a uma consulta. Tão logo a Internet tornou-se popular e os usuários passaram a buscar informações através de serviços como *Gopher* e a *WWW - World Wide Web*, novamente o tráfego aumentou consideravelmente.

A Internet não é controlada por nenhum órgão governamental ou comercial, mas sim por organizações voluntárias que controlam os usuários e os artigos publicados na Internet.

A descrição da arquitetura do protocolo TCP/IP em camadas como as do modelo de referência OSI é uma tarefa difícil e certamente controversa. As camadas OSI foram definidas por pesquisadores ao longo de anos, sempre com o compromisso acadêmico de ser um modelo de referência, enquanto que o protocolo TCP/IP não teve qualquer compromisso que não a funcionalidade. Assim sendo, tentar estabelecer uma relação precisa entre as camadas OSI e TCP/IP é algo praticamente impossível.

O modelo mais aceito para descrever a arquitetura TCP/IP é composto de quatro camadas:

- Acesso à rede ou Interface;
- Internet;
- Transporte;
- Aplicação.

Este modelo é apresentado na Tabela 1, em comparação ao modelo de referência OSI.

Tabela 1 - Comparação entre MR-OSI e MR-TCP/IP

Camadas MR-OSI	Camadas MR-TCP/IP
Aplicação: Aplicações que usam a rede	**Aplicação:** Aplicações que usam a rede
Apresentação: Padronização da representação dos dados	
Sessão: Gerência do diálogo entre aplicações	
Transporte: Transporte fim a fim com correção de erros	**Transporte:** Transporte de dados fim-a-fim
Rede: Transferência de pacotes na rede	**Inter-rede:** Roteamento de datagramas
Enlace: Comunicação confiável ponto-a-ponto	**Acesso ao Meio:** Acesso ao nível físico
Físico: Características físicas da rede	

Os principais protocolos do modelo TCP/IP são os que compõem o próprio nome do modelo, o protocolo do nível de transporte TCP e o protocolo do nível de rede IP. O TCP (*Transmission Control Protocol*) Protocolo de Controle de Transferência é um protocolo de transporte confiável e orientado à conexão definido nos RFCs 793, 1122, 1323, 2018 e 2581. O TCP fornece multiplexação e demultiplexação, detecção de erros e transferência de dados *full-duplex*. É um protocolo fim-a-fim, ou seja, interpretado apenas pela origem e pelo destino da comunicação. O IP (*Internet Protocol*) é responsável pelo endereçamento lógico da Internet e pelas informações de roteamento passadas aos roteadores para que estes tomem decisões. A versão mais utilizada na internet é o IPv4 definido na RFC 791.

3. SEGURANÇA EM REDES DE COMPUTADORES

Um sistema computacional seguro é definido por Garfinkel e Spafford como sendo “aquele que se comporta de maneira esperada”.(1996). A observação do comportamento esperado em comparação com o comportamento apresentado, pode ser entendida como o nível de confiança do sistema e indica o quanto podemos confiar no seu funcionamento. O comportamento esperado é formalizado dentro da política de segurança do sistema e regula as metas que este deve cumprir. Desta forma, um evento de segurança normalmente está relacionado a uma violação de normas ou procedimentos que dependem do sistema em uso.

Outra definição de segurança de computadores baseia-se na confidencialidade, na integridade e na disponibilidade dos recursos do sistema. Confidencialidade significa que a informação deve estar acessível somente às pessoas autorizadas. Integridade requer que a informação permaneça intacta e inalterada por acidentes ou ataques. Disponibilidade significa que os recursos computacionais estejam funcionando adequadamente, sem degradação de acesso, sempre que usuários autorizados necessitarem dos seus recursos. Um sistema seguro deve proteger seus dados e recursos de acesso não autorizado, de intromissão ou de bloqueio de uso.

Com o aparecimento dos sistemas computacionais compartilhados e multiusuários, já havia uma preocupação em prover algum mecanismo de controle de acesso como a primeira linha de defesa (LAMPSON, 1974). Entretanto isto somente limitava o acesso aos objetos do sistema, mas não restringia o que pode ser feito com ou pelos objetos em si (DENNING, 1987). O controle de acesso não modela e não pode prevenir fluxo de informação não autorizado através do sistema. Alem disso, existem sistemas em que o controle de acesso é arbitrário e a responsabilidade de proteção dos dados depende de ações do usuário final. Isto freqüentemente exige que o usuário

entenda os mecanismos de proteção oferecidos pelos sistemas e como é possível obter a segurança desejada utilizando estes mecanismos.

Sistemas com mecanismos muito rígidos de segurança restringem operações de leitura e escrita. A facilidade de conectividade de computadores é inversamente proporcional à segurança. Desta forma, um sistema completamente seguro pode ser pouco útil.

Podem ser implementados modelos de proteção e controles de acesso para evitar que atacantes externos consigam penetrar nos sistemas, mas estes controles não serão muito úteis contra atacantes internos ou contra o comprometimento do módulo de autenticação. Se uma senha de acesso a uma conta do sistema é fraca, e foi comprometida, a medida de controle não tem como prevenir a perda ou corrupção dos dados aos quais aquela conta tem acesso (CANSIAN, 1997). Em geral, métodos estáticos de proteção podem simplesmente ser insuficiente ou podem tornar o sistema extremamente restritivo aos seus próprios usuários.

Deve-se considerar ainda, a extrema dificuldade em desenvolver softwares complexos, funcionais e isentos de erros. Diversas falhas em sistemas freqüentemente se manifestam como vulnerabilidades na segurança.

Métodos estáticos e complexos de segurança têm sérios problemas. O uso de métodos dinâmicos de segurança como verificadores de comportamento são mais indicados para detectar e proteger falhas de segurança e detectar intrusos e intrusões. Os sistemas de detecção de intrusão executam a tarefa de ser a última linha de defesa dentro do esquema de proteção.

Muitas empresas já sentiram os benefícios do uso de redes de computadores: processos internos mais rápidos, comunicações dinâmicas, produtividade incrementada, conquista de um mercado global, etc. As tendências levam a um desenvolvimento espantoso das redes como ferramentas de negócios e pessoais.

Quanto mais complexa se torna à rede, maior é o desafio para mantê-la segura. Com a expansão contínua da infra-estrutura da Internet e da computação móvel, multiplicam-se os pontos de acesso a dados corporativos através da Internet e linhas de telefone dial-up. Cada ponto de acesso representa uma possível vulnerabilidade que pode ser aproveitada para conseguir acesso não autorizado à sua rede. Conforme as

estatísticas apresentadas pelo CERT – Computer Emergency Response Team[1], a respeito apenas dos incidentes de segurança reportados nos últimos anos (Figura 3), há a confirmação do aumento significativo, ano após ano, com exceção de 1996 para 1997, de ataques devido ao crescimento e popularização da Internet.

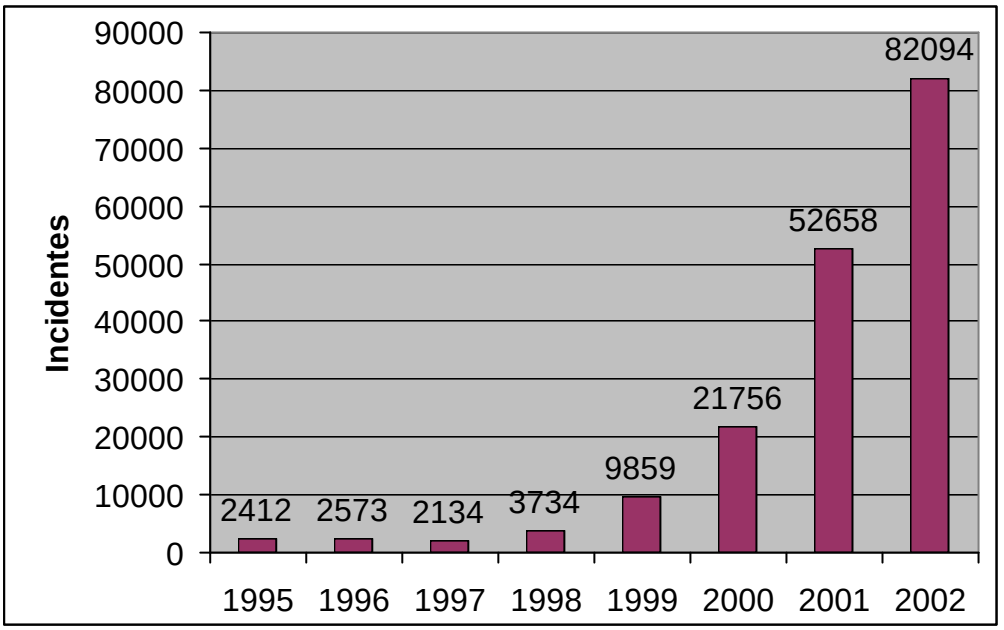

Figura 3 - Incidentes reportados entre 1995 e 2002 ao CERT[2]

Atualmente, o crescimento da interatividade nos mais diferentes ramos de atividade, gera uma demanda elevada em relação às facilidades de acessar e compartilhar informações entre grupos de trabalho, muitas vezes envolvidos em projetos conjuntos. Esta facilidade deve ser transparente o suficiente para permitir agilidade e em contra partida oferecer segurança para evitar roubo, cópia ilícita ou alterações indevidas de informações.

Proteger a propriedade intelectual e ao mesmo tempo permitir acesso transparente ao sistema por pessoal autorizado é um grande dilema. Os administradores de sistemas estão adotando posturas controladoras, optando por estruturar documentos que apresentam de forma explícita a posição da empresa perante seus dados, descrevendo como será o controle, o nível de segurança entre outros detalhes cruciais para total comprometimento do usuário com o sistema.

[1] CERT – Computer Emergency Response Team; Disponível em <http://www.cert.org>
[2] Disponível em <http://www.cert.org/stats/cert_stats.html>; Acessado em jan. 2003

Uma política de segurança deve ser formulada, como primeiro passo, identificando os principais recursos a serem protegidos, definindo quais usuários terão acesso a estes recursos e quais seus privilégios. O estudo para criação desta política de ajudará a estabelecer quais são os objetivos de segurança e a fazer um plano para administrá-los. Deve se conceber uma estratégia bem acabada que reúna as quatro categorias de proteção da informação:

- **Avaliar** vulnerabilidades e assegurar o cumprimento da política de segurança.
- **Proteger** o sistema de informações críticas.
- **Habilitar** o uso seguro da Internet.
- **Gerenciar** e administrar usuários e recursos.

Segundo Campello(2001), "o objetivo da segurança em computadores é dotar os sistemas computacionais de características que impeçam o acesso ou manipulação, intencional ou não, de informações ou recursos por elementos não autorizados".

Estas características são (CAMPELLO,2001):

- **Confiabilidade**: É definida como sendo a capacidade que um sistema tem em responder a uma dada especificação dentro de condições definidas e durante um certo tempo.
- **Integridade**: É a impossibilidade da modificação do estado, das informações e dos recursos do sistema por elementos não autorizados.
- **Disponibilidade**: É a probabilidade de que o sistema esteja funcionando em um dado instante.
- **Autenticidade**: É a possibilidade de identificar a autoria de determinada ação.

3.1. Política de segurança

Uma política de segurança é um conjunto de leis, regras e práticas que regulam como uma organização gerencia, protege e distribui suas informações e recursos(SOARES, 1995).

O Documento gerado contendo de forma explícita as regras e posturas controladoras que devem ser seguidas pelos usuários da rede deve conter informações descrevendo como será o controle de acesso aos recursos e informações, os níveis de segurança, as atividades permitidas e as negadas, e vários outros detalhes respectivos para que o usuário tenha conhecimento e comprometimento, sabendo como deve se comportar.

Este documento deve definir o que é permitido e o que é proibido em um sistema (GIL, 1994), ou seja, a expressão formal das regras pelas quais é fornecido acesso aos recursos tecnológicos da empresa.

3.2. Formas de Segurança

Existem diversos modelos de segurança e algumas classificações a respeito das políticas de seguranças adotadas pelas empresas.

Segundo Chapman e Zwicky (2000), são definidas as seguintes formas de proteção:

- **Nenhuma segurança**: esta é a forma de segurança mais simples e é representada pela ausência de investimento em qualquer recurso de segurança;
- **Segurança por obscuridade**: neste modelo, um sistema é presumido seguro, simplesmente devido ao fato de ninguém, supostamente, ter conhecimento sobre os métodos de operação, o conteúdo, a existência e outros aspectos pertinentes ao sistema;
- **Segurança baseada em máquina**: nesta abordagem, todos os esforços de segurança são concentrados separadamente em cada máquina. Faz-se todo o esforço para evitar ou amenizar problemas de segurança que poderiam afetar uma máquina em particular;
- **Segurança baseada em rede**: esta abordagem apresenta os esforços de segurança voltados para a rede, controlando o acesso às máquinas da rede e também os serviços por ela oferecidos.

Há ainda uma abordagem, mais simplificada, definindo duas filosofias para sustentar as políticas de segurança:

?? **Proibitiva**: onde tudo que não é expressamente permitido será proibido;

?? **Permissiva**: onde tudo que não é expressamente proibido será permitido.

As decisões tomadas pelos responsáveis pela rede, relacionadas à segurança, irão determinar o quanto esta rede é segura ou insegura, qual o nível de facilidade em utilizá-la e quais funcionalidades serão oferecidas por ela.

3.3. Objetivos da Segurança

Para que sejam tomas decisões corretas em relação a segurança, é necessário determinar quais são as metas de segurança e qual o comprometimento efetivo devido as ações adotadas. Deve-se observar as seguintes condições:

?? **Serviços oferecidos versus segurança oferecida:** os serviços oferecidos aos usuários carregam seus próprios riscos de segurança. Para alguns serviços, os riscos apresentados são superiores aos benefícios oferecidos, cabendo ao administrador fazer a opção de eliminar o serviço ou mantê-lo e procurar alternativas para torná-lo mais seguro.

?? **Finalidade de uso versus segurança**: sistemas preocupados com a facilidade de uso, normalmente deixam de controlar o acesso de usuários aos seus recursos, pedindo quando muito, uma senha inicial de acesso, comum a todos os usuários, relegando assim a preocupação com a segurança. A solicitação de senhas individuais e o controle detalhado dos poderes de acesso aos recursos para cada usuário tornam o sistema menos conveniente, porém aumentam a sua segurança.

?? **Custo de segurança versus risco**: A implantação de recursos de segurança nas redes de computadores geram dois tipos de custos. O custo monetário, causado pela aquisição, implantação e manutenção de hardware e software destinado a esta tarefa. O segundo, o custo de desempenho, aumentados devido a utilização do poder de processamento dos equipamentos pelos sistemas de cifragem, e pelos algoritmos complexos

de proteção implementados nos sistemas. Deve-se analisar, também, os riscos da perda de privacidade e as possibilidades de perder informações.

3.4. Incidentes de Segurança

Incidentes de segurança são definidos como sendo “Qualquer evento real ou suspeito adverso em relação à segurança de computadores. Exemplos disto são: intrusões de sistemas computacionais pela rede, ocorrências de vírus de computador ou sondagens de vulnerabilidades através da rede para alcançar sistemas computacionais” (WEST-BROWN , 98).

Os incidentes de segurança podem ser divididos em eventos, ameaças, ataques e vulnerabilidades.

3.4.1. Eventos

Um evento é uma ação dirigida a um objetivo com a intenção de mudar seu estado. Um evento de segurança envolve uma violação ou quebra da política de segurança do sistema. A perda de confiabilidade dos dados, a quebra de sigilo, o rompimento da integridade das informações ou dos recursos do sistema ou a indisponibilidade de um recurso ou informação são exemplos de eventos de segurança.

Howard (1998), classifica diversas ações desencadeadas por atacantes com o propósito de atingir seus objetivos ou alvos:

Tabela 2 - Ações e Alvos descritos por Howard (1998)

Ações	Alvos
Sondagem (*probe*)	Conta de usuário
Varredura (*scan)*	Processo
Inundação (*flood*)	Dado
Autenticação (*authenticate*)	Componente
Desvio (*bypass*)	Computador
Máscara (*spoof*)	Rede
Leitura (*read*)	Inter-rede
Cópia (*copy*)	
Roubo (*steal*)	
Alteração (*modify*)	
Destruição (*delete*)	

Fonte: Howard e Longstaff, 1998

3.4.2. Ameaças

Todo sistema está sujeito a diferentes tipos de ameaças, sejam elas internas ou externas, acidentais ou maliciosas. Segundo Soares (1995), uma ameaça consiste na possibilidade de violação da segurança de um sistema. São algumas ameaças aos sistemas computacionais:

?? destruição de informações ou de outros recursos;

?? modificação ou deturpação de informações;

?? roubo, remoção ou perda de informações ou de outros recursos;

?? revelação de informações;

?? interrupção de serviços.

A concretização de alguma dessas ameaças, ocasionada por uma ação intencional, configura um ataque.

Howard (1998), propõe uma taxonomia, mostrada na figura abaixo, que caracteriza bem os diferentes tipos de incidentes de segurança.

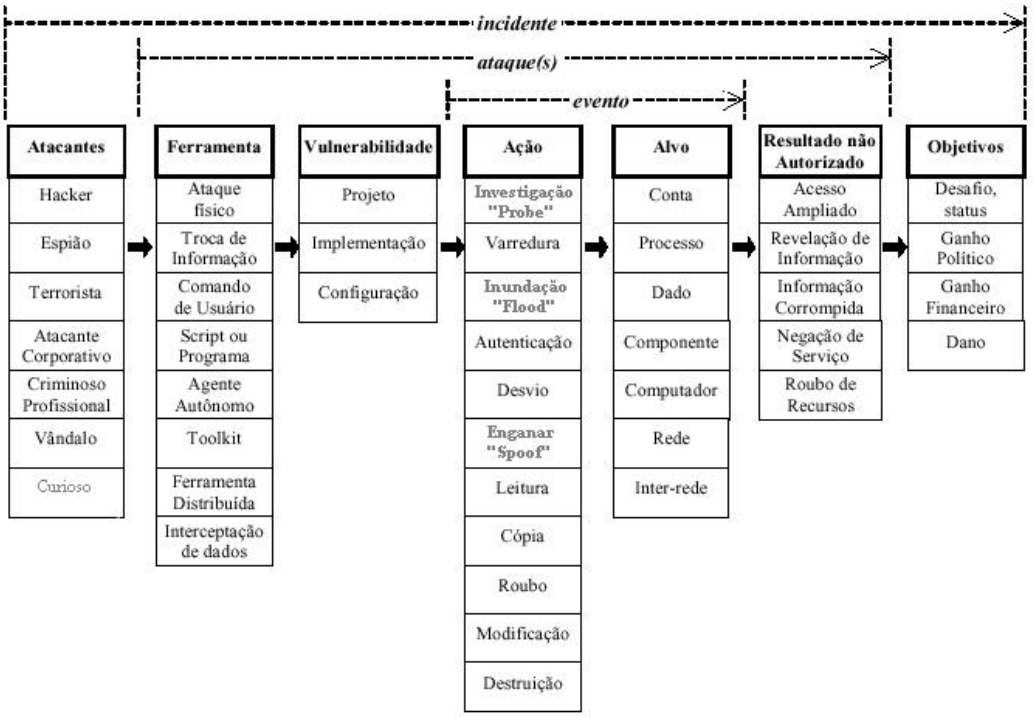

Figura 4 - Taxonomia de incidentes de ataque

3.4.3. Ataques

Em algumas vezes, um evento que ocorre em um computador ou em uma rede, é parte de uma série de passos que resultarão em um acontecimento não autorizado. Este evento passa a ser considerado como parte de um ataque. Um ataque tem vários elementos. Primeiro, é criada uma série de passos tomados por um atacante. Entre estes passos, estão algumas ações direcionadas ao alvo (um evento), bem como o uso de algumas ferramentas para explorar uma vulnerabilidade. Segundo, um ataque é direcionado para alcançar um resultado não autorizado do ponto de vista da empresa ou pela política de segurança adotada. Finalmente, um ataque é uma série de passos intencionais iniciados por um atacante. Isto difere um ataque de uma ameaça.

Um ataque é definido como uma série de atitudes tomadas intencionalmente por um atacante para alcançar um resultado não autorizado.

Uma outra definição é a de que um ataque é uma agressão à segurança do sistema derivada de uma ameaça inteligente, uma tentativa deliberada para evadir serviços de segurança e violar a política de segurança de um sistema. (RFC-2828)

Alguns ataques conhecidos:

?? **Furto e quebra de senha**: O atacante invade um servidor e rouba (copia) o arquivo de senhas, este arquivo é submetido a um software que fará a quebra (decodificação) das senhas, permitindo acesso não autorizado às contas destes usuários.

?? **Engenharia Social**: Engenharia Social é o termo utilizado para a obtenção de informações importantes de um sistema, como a senha de um usuário, através da confiança. Os ataques por engenharia social são antigos, porém ainda surtem efeito. Este tipo de ataque consiste basicamente em mentir, contar uma estória a um usuário inocente, visando obter informações confidenciais ou ainda, convencer o usuário a executar código malicioso em seu sistema.

?? **Denial of Service (DoS)**: O ataque de negação de serviço consiste em sobrecarregar um servidor com uma quantidade excessiva de solicitações de serviços até que este servidor seja reinicializado ou tenha sua

performance comprometida. Existe ainda, ataques DoS distribuídos onde o atacante invade diversos *host* instalando softwares invasores que a partir de comandos dados pelo atacante parte ao mesmo tempo ataques sobre um servidor tirando-o do ar.

?? **Falhas de Autenticação**: Explorando vulnerabilidades dos sistemas de autenticação, o invasor entra no host sem a necessidade de preencher informações como nome de usuário de senha.

?? **Falhas de protocolo**: Beneficiando-se de vulnerabilidades em protocolos de comunicação o atacante gera ataque DoS (Deny of Service) negação de serviços para conseguir prejudicar a performance de algum *host* ou mesmo tirá-lo de ação.

?? **Spoofing**: É o ato de usar um *host* para personificar outro. Isto é feito forjando o endereço de origem de um *host* empenhado na autenticação de máquinas individualmente. Normalmente o atacante gera um ataque DoS sobre um *host* e então faz com que sua máquina passe a responder pelo *host*.

?? **Sniffing**: O *sniffer* é um programa que analisa o tráfego de rede, geralmente utilizados para gerenciamento de redes, mas podem ser utilizados por atacantes com o intuito de descobrir informações relevantes sobre aquela rede monitorada.

?? **Scanners de porta**: Os *scanners* são software que varrem a rede em busca de portas TCP abertas e, portanto vulneráveis, onde seja possível efetuar uma invasão.

?? **Trojan** ou **Cavalo de Tróia**: O termo vem da mitologia grega que relata uma passagem onde os gregos deram aos troianos um cavalo de madeira gigante aparentemente como oferta de uma proposta de paz. Porém, após os troianos recolherem o cavalo para dentro das paredes que protegiam a cidade, soldados gregos saíram de dentro da barriga oca do cavalo, durante a noite e abriram as portas da cidade aos invasores. Por analogia, o termo

trojan é utilizado quando programas destrutivos são inseridos ou mascarados em programas ou aplicativos benignos.

3.4.4. Vulnerabilidades

A maioria dos desenvolvedores de software, não tem preocupações específicas com segurança, e mesmo aqueles que tomam os cuidados devidos, sabem que é praticamente impossível desenvolver aplicações completamente seguras, mesmo tomando cuidado com as possíveis falhas de segurança que o sistema poderá vir a provocar. A maioria dos sistemas possui algum nível de vulnerabilidade.

"A vulnerabilidade é uma fragilidade de um sistema, permitindo uma ação não autorizada" (HOWARD, 1998).

Howard definiu três categorias de vulnerabilidade:

?? **Vulnerabilidade de projeto**: é uma vulnerabilidade inerente ao projeto ou especificação de hardware ou software, por meio de qual, mesmo em uma implementação perfeita resultará em uma vulnerabilidade.

?? **Vulnerabilidade de implementação**: é uma vulnerabilidade resultante de um erro gerado na implementação do software ou hardware. Uma falha de implementação ainda que o projeto tenha sido satisfatório.

?? **Vulnerabilidade de configuração**: é uma vulnerabilidade resultante de um erro na configuração de um sistema, como, por exemplo, contas de sistema com senha padrão, permissão de escrita livre para novos arquivos, ou habilitar serviços contendo vulnerabilidades.

3.5. Segurança no TCP/IP

Uma das principais deficiências no aspecto de segurança do protocolo IP é a incapacidade deste, de autenticar uma máquina na rede. Em outras palavras, com base no endereço IP de origem de um pacote recebido, é impossível determinar com certeza a identidade da máquina que o tenha originado. Há também poucas garantias de que o conteúdo de um pacote recebido não tenha sido alterado, muito menos ainda que a privacidade dos dados contidos tenha sido preservada.

Os ataques que exploram tal falha têm como tática mais comum, a personificação de uma máquina na rede. Sua finalidade é a de obter informações sigilosas como senhas, abusar da confiança que máquinas mantém entre si, até ações mais sutis e sofisticadas, como alterar o conteúdo dos dados que estejam de passagem para outros destinos.

3.6. Criptografia

A criptografia vem, na sua origem, da fusão de duas palavras gregas: CRIPTO = ocultar, esconder GRAFIA = escrever. Criptografia é arte ou ciência de escrever em cifra ou em códigos. É então um conjunto de técnicas que tornam uma mensagem incompreensível permitindo apenas que o destinatário que conheça a chave de criptografia possa decifrar e ler a mensagem com clareza.

A criptografia é um método de cifrar mensagens para que estas possam trafegar em meio público sem serem compreendidas por terceiros. É uma técnica muito antiga, e durante a segunda guerra mundial foi muito utilizada para transmitir mensagens no campo de batalha, a partir daí começou a ser muito pesquisada e passou a ser usada para proteger mensagens nos meios de comunicação. Existem hoje, basicamente duas técnicas importantes de criptografia para suprir o problema de segurança eletrônica de mensagens, ambas baseadas na utilização de chaves de criptografia, que são usadas para fazer a codificação e decodificação das mensagens. O método de chaves simétricas e o método de chaves assimétricas.

3.6.1. Criptografia de chaves simétricas

Esse método (Figura 5) é conhecido normalmente como criptografia tradicional, funciona muito bem em aplicações limitadas, onde o remetente e o destinatário se preparam antecipadamente para o uso da chave. Para o funcionamento correto deste método, todas as pessoas envolvidas devem conhecer a chave, pois quando uma mensagem cifrada chega ao destino, ela só poderá ser decifrada por quem possuir a mesma chave que a cifrou. Este método torna-se pouco eficiente em conexões inseguras, no entanto, quando for utilizado sobre conexões seguras, a criptografia simétrica se torna bastante eficiente.

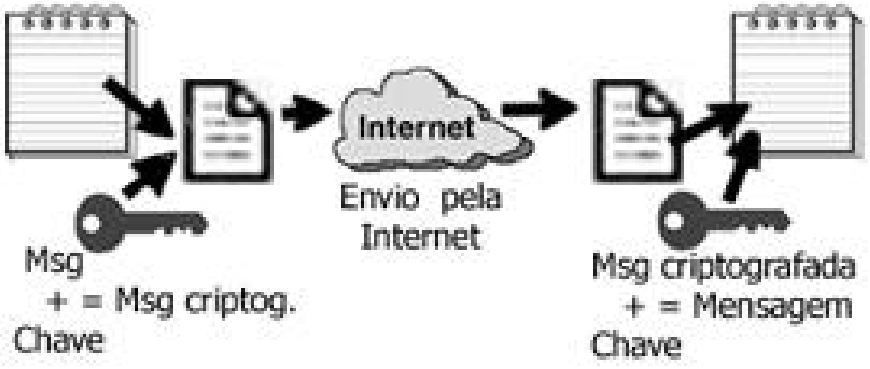

Figura 5 - Esquema de criptografia de chave simétrica

Existem vários algoritmos de chaves simétricas, entre eles se destacam os seguintes:

?? DES – Data Encryption Standard: utiliza uma chave de 56 bits e blocos de 64 bits. Muito utilizado em conexões web seguras com o SSL – Security Socket Layer,

?? IDEA – International Data Encryption Algorithm

?? RC2 e RC4 – algoritmo de criptografia mais rápido que o DES, e pode ter sua confiabilidade incrementada aumentando o tamanho da chave.

3.6.2. Criptografia de Chaves Assimétricas

A criptografia de chave pública ou criptografia assimétrica foi desenvolvida em 1970 e funciona com uma chave para cifrar as mensagens e outra chave para decifrá-las. No sistema de chave pública, cada usuário deve ter duas chaves, uma disponível publicamente e outra secreta. Este método foi patenteado pela RSADSI (RSA Data Security Incorporated). A figura abaixo apresenta o funcionamento da criptografia assimétrica.

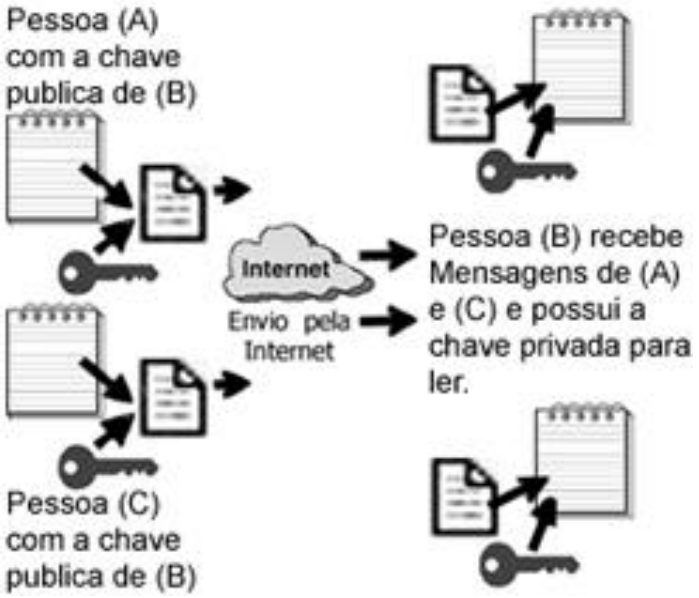

Figura 6 - Criptografia de Chaves Assimétricas

A pessoa A, primeiro obtém a chave pública da pessoa B, então cifra a mensagem usando esta chave para após, enviá-la a pessoa B. A pessoa B, ao receber a mensagem cifrada, irá decifrá-la utilizando sua própria chave privada.

3.6.3. Assinatura Digital

A assinatura digital busca resolver dois problemas não garantidos apenas com uso da criptografia para codificar as informações: a Integridade e a Procedência. Ela utiliza uma função chamada *one-way hash function* também conhecida como: *compression function, cryptographic checksum, message digest* ou *fingerprint*. Essa função gera uma cadeia de caracteres única sobre uma informação, se esse valor for o mesmo tanto no remetente quanto no destinatário, significa que essa informação não foi alterada. Mesmo assim isso ainda não garante total integridade, pois a informação pode ter sido alterada no seu envio e um novo *hash* pode ter sido calculado (MEDEIROS, 2001).

Para solucionar esse problema, é utilizada a criptografia assimétrica com a função das chaves num sentido inverso, onde o *hash* é criptografado usando a chave privada do remetente, sendo assim o destinatário de posse da chave pública do remetente poderá decifrar o *hash*. Dessa maneira é possível garantir a procedência, pois somente o remetente possui a chave privada para codificar o *hash* que será aberto pela sua chave

pública. Já o *hash*, gerado a partir da informação original, protegido pela criptografia, garantirá a integridade da informação.

3.6.4. Certificados Digitais

O Certificado Digital, também conhecido como Certificado de Identidade Digital, associa a identidade de um titular a um par de chaves eletrônicas (uma pública e outra privada) que, usadas em conjunto, fornecem a comprovação da identidade. É uma versão eletrônica (digital) de algo parecido a uma Cédula de Identidade, serve como prova de identidade, reconhecida diante de qualquer situação onde seja necessária a comprovação de identidade.(MEDEIROS, 2001).

O Certificado Digital pode ser usado em uma grande variedade de aplicações, como comércio eletrônico, *groupware* (Intranet's e Internet) e transferência eletrônica de fundos. Dessa forma, um cliente que compra em um shopping virtual, utilizando um servidor seguro, solicitará o Certificado de Identidade Digital deste servidor para verificar: a identidade do vendedor e o conteúdo do certificado por ele apresentado. Da mesma forma, o servidor poderá solicitar ao comprador seu Certificado de Identidade Digital, para identificá-lo com segurança e precisão. Caso qualquer um dos dois apresente um Certificado de Identidade Digital adulterado, ele será avisado do fato, e a comunicação com segurança não será estabelecida.

O Certificado de Identidade Digital é emitido e assinado por uma Autoridade Certificadora Digital (*Certificate Authority*). Para tanto, esta autoridade usa as mais avançadas técnicas de criptografia disponíveis e de padrões internacionais (norma ISO X.509 para Certificados Digitais), para a emissão e chancela digital dos Certificados de Identidade Digital.

A seguir a descrição dos três elementos principais dos certificados digitais (MEDEIROS, 2001):

- Informação de atributo: É a informação sobre o objeto que é certificado. No caso de uma pessoa, isto pode incluir seu nome, nacionalidade, endereço e-mail, sua organização e o departamento da organização onde trabalha.

- ?? Chave de informação pública: É a chave pública da entidade certificada. O certificado atua para associar a chave pública à informação de atributo, descrita acima. A chave pública pode ser qualquer chave assimétrica, mas usualmente é uma chave RSA.
- ?? Assinatura da Autoridade em Certificação (CA): A CA assina os dois primeiros elementos e, então, adiciona credibilidade ao certificado. Quem recebe o certificado verifica a assinatura e acreditará na informação de atributo e chave pública associadas se acreditar na Autoridade em Certificação.

Existem diversos protocolos que usam os certificados digitais para comunicações seguras na Internet (MEDEIROS, 2001):

- ?? Secure Socket Layer ou SSL
- ?? Secured Multipurpose Mail Extensions - S/MIME
- ?? Form Signing
- ?? Authenticode / Objectsigning.22

O SSL é talvez a mais difundida aplicação para os certificados digitais e é usado em praticamente todos os *sites* que fazem comércio eletrônico na rede (livrarias, lojas de CD, bancos, etc.). O SSL teve uma primeira fase de adoção onde apenas os servidores estavam identificados com certificados digitais, e assim tínhamos garantido, além da identidade do servidor, o sigilo na sessão. Entretanto, apenas com a chegada dos certificados para os navegadores é que o lado cliente pode contar também com a identificação, eliminando assim a necessidade do uso de senhas e *logins*.

O S/Mime é também *um* protocolo muito popular, pois permite que as mensagens de correio eletrônico trafeguem cifradas e/ou assinadas digitalmente. Desta forma os e-mails não podem ser lidos ou adulterados por terceiros durante o seu trânsito entre a máquina do remetente e a do destinatário. Além disso, o destinatário tem a garantia da identidade de quem enviou o e-mail.

O Form Signing é uma tecnologia que permite que os usuários emitam recibos online com seus certificados digitais. Por exemplo: o usuário acessa o seu *Internet*

Banking e solicita uma transferência de fundos. O sistema do banco, antes de fazer a operação, pede que o usuário assine com seu certificado digital um recibo confirmando a operação. Esse recibo pode ser guardado pelo banco para servir como prova, caso o cliente posteriormente negue ter efetuado a transação.

O *Authenticode* e o *Object Signing* são tecnologias que permitem que um desenvolvedor de programas de computador assine digitalmente seu software. Assim, ao efetuar uma copia (*download*) de um software pela Internet, o usuário tem certeza da identidade do fabricante do programa e que o software se manteve íntegro durante o processo de *download*. Os certificados digitais se dividem em basicamente dois formatos: os certificados de uso geral (que seriam equivalentes a uma carteira de identidade) e os de uso restrito (equivalentes a cartões de banco, carteiras de clube etc.). Os certificados de uso geral são emitidos diretamente para o usuário final, enquanto que os de uso restrito são voltados basicamente para empresas ou governo.

3.6.5. SSL – Secure Sockets Layer

O SSL é um protocolo de segurança projetado pela *Netscape Communications Corporation*, a empresa do famoso navegador Netscape. O SSL destina-se a dar segurança durante a transmissão de dados sensíveis por TCP/IP. O SSL fornece criptografia de dados, autenticação de servidor e integridade de mensagem para transmissão de dados pela Internet. O SSL versão 2.0 suporta apenas autenticação de servidor, ao passo que a versão 3.0 suporta a autenticação tanto de cliente como de servidor.(MEDEIROS, 2001).

Quando o navegador("cliente") conecta-se a uma página protegida por SSL, o servidor do SSL envia uma solicitação para iniciar a sessão segura. Se o navegador suporta SSL, então retorna uma resposta. Durante este *handshake* ("apertar de mãos") inicial, o servidor e o navegador trocam informações seguras. A resposta do navegador define um número único para identificar a sessão, os algoritmos de criptografia e os métodos de compactação que este suporta. Nas informações de segurança fornecidas pelo navegador, o servidor faz sua seleção e a comunica ao navegador. O servidor e o navegador, em seguida, trocam certificados digitais. O servidor também especifica uma chave pública ("chave de sessão") apropriada para o algoritmo de criptografia anteriormente selecionado. O navegador pode, então, usar a chave pública para cifrar

informações enviadas ao servidor, sendo que o servidor pode usar sua chave privada para decifrar essas mensagens. Depois que o servidor e o navegador estão de acordo sobre a organização da segurança, as informações podem ser transmitidas entre os dois, em um modo seguro.

Os dados protegidos pelo protocolo envolvem o uso de criptografia e decriptografia, portanto, o uso do SSL envolve uma carga extra. De fato, o seu uso não apenas aumenta a quantidade de dados transmitidos, mas também cria mais pacotes, tornando mais lenta a transmissão de informações entre o servidor e o browser.

Entretanto, ele pode ser implementado no nível da página da Web. Ou seja, não é necessário implementar proteção do protocolo para cada página de um site na Web que forneça proteção de SSL. O método mais comum de implementação para aplicações de comércio eletrônico é proteger com o SSL apenas aquelas páginas que contêm informações confidenciais e sensíveis, tais como informações pessoais e de cartão de crédito.

Atualmente o mecanismo de criptografia do SSL utiliza chave pública RSA com tamanho de chaves de 128bits para implementar transmissão segura. Quanto maior o número de bits na chave de criptografia, tanto mais difícil será quebrar a chave.

3.7. Firewall

Firewall (parede contra fogo) é um conjunto de sistemas (um ou mais equipamentos/roteadores) situado entre uma rede privada e uma rede externa, que têm como principal função interceptar todo tráfego entre ambos, e com base na política de segurança interna, permitir ou não a sua passagem (RANUM, 1995) através da utilização de filtros.

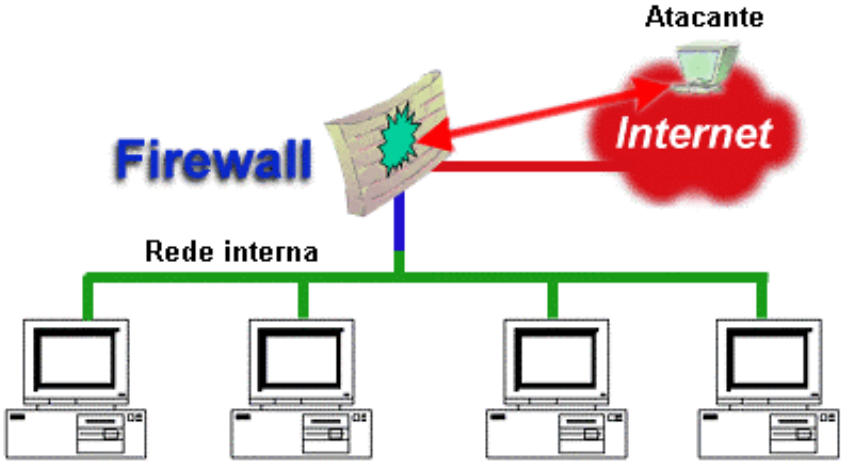

Figura 7 - Firewall

O principal objetivo de um *firewall* é proteger o sistema de ataques originados fora da rede através da implantação de barreiras na fronteira entre a rede interna e a rede externa. Estas barreiras são compostas de filtros e de um direcionador (*gateway* ou *proxy*). Como efeito secundário, ele também pode ser utilizado para regular o uso de recursos externos pelos usuários internos proibindo ou permitindo acesso de dentro para fora da rede de acordo com as características do protocolo a ser utilizado. Entretanto um *firewall* não pode proteger o sistema contra ataques originados por usuários que se encontram dentro da rede.

4. SISTEMAS DE DETECÇÃO DE INTRUSÃO

Uma intrusão é definida por Heady (1990) como “qualquer conjunto de ações que tentam comprometer a integridade, confidencialidade ou disponibilidade de um recurso computacional”.

Desde que a detecção de intrusão começou a ser investigada, muita coisa aconteceu. Muitos sistemas de detecção de intrusão estão disponíveis comercialmente e um grande número de instituições acadêmicas tem realizado pesquisas nesta área.

A finalidade de um IDS ou Sistema de Detecção de Intrusão é monitorar um ambiente de rede em busca de um comportamento anormal ou abusivo.

4.1. Ataques por anomalia

Ataques por anomalia significam atividades anormais que podem indicar uma intrusão. Se a observação das atividades de um usuário desviarem do padrão de comportamento, uma anomalia estará ocorrendo. Um problema com a detecção por anomalia é a probabilidade de acréscimo nos alarmes falsos. Uma atividade não corriqueira, mas legítima poderá vir a ser considerada como anormal. O desafio é desenvolver um modelo de validação de comportamento que possa aceitar novas atividades desconhecidas, porém legítimas.

É difícil construir um protótipo como este pela mesma razão que é difícil construir um sistema compreensível de detecção por abuso (descrito no item 4.2), Não é possível antecipar todas as variações possíveis de comportamento. Esta tarefa pode ser discutida de três maneiras:

1. No lugar de generalizar o uso legítimo, o comportamento de um usuário pode ser modelado. A tarefa de caracterizar um estilo regular de

comportamento de um usuário individual é mais fácil do que tentar fazer para todos os usuários simultaneamente.

2. O modelo de comportamento pode ser aprendido por exemplos de uso legítimo, ao invés de ter que descrever os comportamentos possíveis manualmente.

3. Detectar intrusos em tempo real, enquanto os usuários estiverem digitando comandos, é muito difícil porque a ordem dos comandos podem variar muito. Em muitos casos, é suficiente reconhecer que a distribuição dos comandos sobre uma sessão inteira, ou ainda um dia inteiro, for diferente do normal.

4.2. Ataques por abuso

Ataques por abuso referem-se a ataques conhecidos que exploram vulnerabilidades conhecidas do sistema. Detecção por abuso podem ser muito poderosas naqueles ataques conhecidos e que foram programados nos sistemas de detecção. No entanto, não é possível antecipar todos os diferentes ataques que podem ocorrer e a tarefa de estar sempre atualizando o sistema pode ser muito árdua.

Muitos IDSs são baseados no modelo geral proposto por Denning (1987). Este modelo é independente de plataforma, vulnerabilidade de sistemas e tipos de intrusão. Ele mantém um conjunto de perfis de usuários, conferindo um registro de auditoria com o perfil apropriado, atualizando o perfil quando necessário e relatando qualquer anomalia detectada. Outro componente, um conjunto de regras, é usado para detectar abusos.

Atualmente, os sistemas implementam o modelo geral com técnicas diferentes. Muitas vezes métodos estatísticos são usados para medir quão anômalo é o comportamento, isto é, quanto os comandos usados são diferem do que se espera como comportamento normal. Tal aproximação requer que a distribuição de tais comportamentos seja conhecida. O comportamento pode ser representado como um modelo baseado em regras (LUNT, 1993), geração de um modelo de previsão ou usando análise de transição de estado (PORRAS et al. 1995). Técnicas de comparação de padrões são usadas para determinar se a seqüência de eventos é parte de um

comportamento normal, constitui uma anomalia ou bate com a descrição de um ataque conhecido.

Os Sistemas de Detecção de Intrusão também diferem quanto a serem On-line ou Off-line. IDSs Off-line são executados periodicamente e detectam intrusões após o fato, verificando em registros dos sistemas. IDSs On-line são projetados para detectar intrusões enquanto elas estão acontecendo, permitindo assim uma rápida intervenção. IDSs On-line são computacionalmente muito caros porque requerem monitoramento contínuo. Decisões precisam ser tomadas rapidamente com menos dados não sendo assim muito confiáveis.

Há muitas pesquisas propondo Sistemas de Detecção de Intrusão que empregam redes neurais para detecção on-line de intrusos (DEBAR et al. 1992). Estes sistemas aprendem a prever o próximo comando baseado na seqüência prévia de comandos de um usuário específico. Através de uma janela de transferência, a rede recebe os ?mais recentes comandos como entrada. A rede é recorrente, isto é, parte da saída é recolocada como entrada para o próximo passo. Então a rede está constantemente observando uma nova direção e "esquecendo" comportamentos antigos. O tamanho da janela é um parâmetro importante: Se for muito pequeno, então teremos muitos falso-positivos, mas se for muito grande, a rede pode não generalizar bem as novas seqüências. Alguns destes sistemas (DEBAR et al. 1992) podem prever o próximo comando corretamente em torno de 80% das vezes e aceita um comando como previsível (90 % das vezes).

Alarmes falsos de IDS são problemáticos, quando uma atividade normal for considerada como um ataque hostil, administradores de sistemas terão que gastar seu tempo verificando-os. Estes falso ataques causam prejuízos financeiros nas corporações quando o acesso a recursos técnicos são negados ou recursos de segurança são redirecionados para investigar eventos não intrusivos.

4.3. Características desejáveis em um Sistema de Detecção de Intrusão

As seguintes características, baseadas em uma lista fornecida por Crosbie e Spafford (1995) são desejáveis em um Sistema de Detecção de Intrusão:

?? Um IDS deve estar em execução contínua com um mínimo de supervisão humana

?? Um IDS deve ser tolerante a falhas:

- Um IDS deve estar apto a se recuperar de uma falha no sistema, tanto acidentais quanto as causadas por atividades maliciosas.
- Após uma falha no sistema, um IDS deve estar apto a recuperar o estado imediatamente anterior e dar continuidade as operações não afetadas

?? Um IDS deve resistir a subversão:

- Deve possuir um nível considerável de dificuldade para que um atacante possa desabilitá-lo ou modificá-lo.
- O IDS deve estar apto a monitorar-se e detectar se foi modificado por um atacante.

?? O IDS deve impor uma sobrecarga mínima ao sistema onde estiver rodando e interferir o mínimo possível em seu funcionamento normal.

?? Ele deve ser configurável para implementar corretamente as políticas de segurança dos sistemas aos quais está monitorando.

?? Deve ser fácil de implantar. Isto pode ser alcançado através da portabilidade para arquiteturas e sistemas operacionais diferentes, através de um único mecanismo de instalação e ser de fácil compreensão e uso pelo operador.

?? O IDS deve ser adaptável a mudanças no sistema e comportamento dos usuários a qualquer tempo. Por exemplo, novas aplicações que foram instaladas, usuários que mudaram de setores ou atividades ou novos recursos disponíveis que podem causar alterações nos padrões do sistema.

?? Ele deve estar apto a detectar ataques:

- O IDS não deve anunciar qualquer atividade legítima como sendo um ataque. (falso positivo).
- O IDS não deve falhar ao anunciar qualquer ataque real . (falso negativo) Deve dificultar ao atacante mascarar suas ações para que possa detecta-lo.

- O IDS deve reportar as intrusões tão logo aconteçam.
- O IDS deve ser o mais genérico possível para detectar diferentes tipos de ataques.

4.4. Problemas comuns em Sistemas de Detecção de Intrusão

Muitos dos IDS existentes, sofrem de pelo menos dois dos seguintes problemas:

Primeiro, as informações utilizadas pelos sistemas de detecção de intrusão são obtidas de registros de auditoria ou de pacotes que trafegam na rede. Os dados tem que atravessar um longo caminho da origem até o IDS, e neste processo, tem grande possibilidade de serem destruídos ou modificados por um atacante.

Segundo, os IDS usam continuamente recursos adicionais no sistema que estão monitorando mesmo quando não há intrusões acontecendo, por que os componentes de detecção de intrusos tem que estar funcionando todo o tempo.

Terceiro, pelos componentes dos sistemas de detecção de intrusão serem implementados separadamente, eles estão mais suscetíveis a alterações. Um intruso pode desabilitar ou modificar os programas que estão rodando em um sistema tornando o IDS inconfiável ou fora de uso.

4.5. Classificação dos Sistemas de Detecção de Intrusão

Os sistemas de detecção de intrusão podem ser classificados em três categorias principais:

4.5.1. Quanto ao momento do ataque

Podemos caracterizar um IDS quanto ao momento da detecção de um ataque, o IDS pode enviar um alerta:

- ?? Antes que o ataque aconteça;
- ?? Enquanto o ataque está acontecendo;
- ?? Depois que o ataque aconteceu;

4.5.2. Quanto a tecnologia do analisador de eventos:

Podemos também caracterizá-los quanto à tecnologia empregada no analisador de eventos. Um IDS pode trabalhar com:

?? Análise de assinaturas;

?? Análise estatística;

?? Sistemas adaptativos;

4.5.3. Quanto ao sistema que está monitorando ou agindo

A divisão clássica, sugerida pela ICSA[3], é quanto ao sistema que o IDS está inserido e monitorando, desta forma podemos ter:

?? IDS baseado em rede;

O *network*-based *IDS* (ou NIDS) é o modelo IDS mais comum e no qual muitas pessoas pensam quando falam em detecção de intrusão. O NIDS vigia o tráfego da rede e analisa os pacotes capturados em busca de ataques. Essencialmente, ele compara o tráfego capturado ao seu repertório interno de "modelos" de invasões conhecidos. Quando um pacote ou uma série de pacotes percorre o fio e combina com um dos tipos de invasão predefinidos, o sistema dá o alarme, faz o registro, evita o ataque e assim por diante. Apesar dos dois modelos estarem começando a se entrecruzar, cada um é capaz de detectar quebras de segurança que o outro não consegue. Os ataques amplos a portas podem ser pegos por dispositivos NIDS, mas um arquivo */etc/shadow* modificado só é pego por um módulo baseado em host, por exemplo.

?? IDS baseado em host;

Os modelos baseados em host, em geral, requerem que um agente específico da plataforma seja implementado em cada alvo vigiado. Embora alguns produtos baseados em host estejam começando a vigiar os ataques a rede, eles são mais focados em monitorar logs, observar processos e executar checagem de integridade binária.

?? Verificador de integridade de arquivos;

---

[3] Disponível em <http://www.icsalabs.com/html/communities/ids/index.shtml>

4.6. Padronizações de Sistemas de Detecção de Intrusão

A maioria dos Sistemas de Detecção de Intrusão é composta por vários módulos e componentes os quais tem necessidade de comunicar-se entre si. Esta necessidade fez com que surgissem alguns esforços para a padronização destes sistemas. O IETF – Internet Engineering Task Force, criou um grupo chamado IDWG – Intrusion Detection Working Group, com a finalidade de criar padrões para os Sistemas de Detecção de Intrusos. Até agora, existem dois padrões criados por este grupo, o CIDF – Common Intrusion Detection Framework e o CISL – Common Intrusion Specification Language

4.6.1. CIDF – Common Intrusion Detection Framework

O Common Intrusion Detection Framework – CIDF[4] (STANIFORD-CHEN, 1998) é um esforço para desenvolver protocolos e interfaces de programação de aplicações para que os projetos de pesquisa de detecção de intrusão possam compartilhar informações e recursos e para que os componentes de IDS possam ser reusados por outros sistemas. Este esforço, foi iniciado por Teresa Lunt enquanto ela estava no ITO – Information Technology Office do DARPA[5] como parte do programa "*Information Survivability*" com o foco de permitir aos projetos do DARPA trabalharem juntos. O CIDF é um modelo conceitual e propõe o agrupamento de um conjunto de componentes que definem uma ferramenta IDS (STANIFORD-CHEN, 1998):

?? Gerador de Eventos – E-box - componente para obter eventos a partir de um meio externo ao CIDF, gera os eventos mas não os processa. Pode ser um simples monitor de rede para gerar eventos baseados em análises de tráfego, através de filtros.

?? Analisador de Eventos – A-box – componente, utilizando regras previamente definidas, analisa os eventos coletados, gerando um resumo dos dados coletados

?? Base de Dados de Eventos – D-box – componente para armazenar os eventos e resultados analisados para uso futuro.

[4] CIDF-Common Intrusion Detection Framework. Disponível em < http://www.isi.edu/gost/cidf/>
[5] DARPA – Defense Advanced Research Project Agency

?? Unidade de Resposta – R-box – componente responsável pelas ações que podem compreender o encerramento de processos, reinicialização de conexões, alterações de permissões de arquivos, notificação à gerência, entre outros.

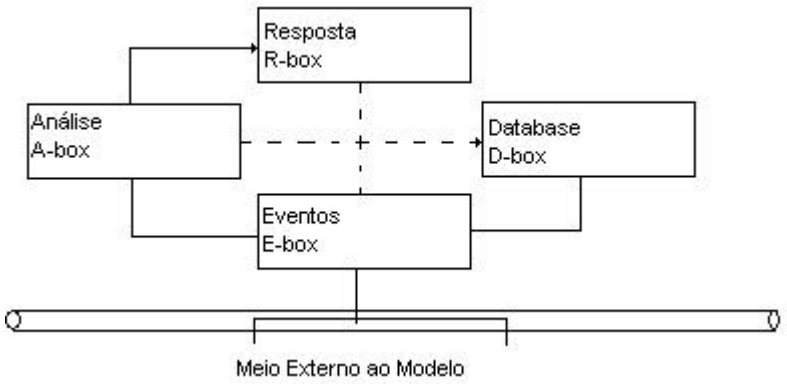

Figura 8 - Modelo Conceitual CIDF <http://www.isi.edu/gost/cidf/>

Todos estes componentes devem ter as seguintes características:

?? reutilização em um contexto diferente do qual foram originalmente desenvolvidos;

?? Deve ser elaborados em módulos diferentes com funções distintas, compartilhando informações;

?? Deve possibilitar a identificação de novos componentes pelos demais componentes;

4.6.2. CISL – Common Intrusion Specification Language

A Common Intrusion Specification Language - CISL é uma tentativa de padronização de transferência de informações e interoperabilidade entre Sistemas de Detecção de Intrusão e foi descrita para ser usada para disseminar registro de eventos, resultado de análises entre componentes e sensores de IDS com sua estação de gerenciamento. Expressar informações sobre eventos, ataques e respostas entre estes sistemas de detecção de intrusão. Foi projetada para ser flexível e eficiente para beneficiar o programador das aplicações. Em troca destes benefícios, é preciso esforçar-se para assegurar que a linguagem não esteja carregada de ambigüidades perigosas.

A CISL utiliza uma sintaxe chamada S-expression, ou seja, agrupamento de átomos usando parênteses. Estas expressões agrupam recursivamente indicadores e dados e são compostas por:

?? SID – Identificador Semântico

?? Dado – Informação relacionada ao SID

As expressões da linguagem Lisp são exemplos de S-expression. No quadro abaixo, é apresentado um exemplo que ilustra os princípios básicos da CISL.

```
(Delete
    (When
        (Time '12:24 15 Mar 1999 UTC')
    )
    (Initiator
        (UserName 'joe')
        (UserID 1234)
        (HostName 'foo.example.com')
  )
  (FileSource
    (FullPathName '/etc/passwd')
    (HostName 'foo.example.com')
  )
)
```

4.6.3. IAP – Internert Intrusion Alert

O IAP é um protocolo ao nível de aplicação para troca de dados entre agentes IDS sobre TCP e foi proposto por um grupo do IETF. É destinado a transmissão de dados do sensor para a estação de gerenciamento para que esta informe a ocorrência, registre o evento e tome as medidas necessárias.

4.7. SNORT – Um Sistema Peso Leve de Detecção de Intrusão

O Snort é um Sistema de Detecção de Intrusão baseado em rede para ser executado em equipamentos de pequeno porte sobre ambiente operacional Linux. Desenvolvido por Martin Roesch (1999) sob licença GNU/GPL – General Public License (GNU) e em constante desenvolvimento na comunidade Internet, concebido para detectar uma variedade muito grande de tráfego suspeito de rede para pequenas e

médias redes que utilizam o protocolo TCP/IP e sejam compatíveis com a biblioteca *libcap* (JACOBSON et al. 1994). O Snort fornece aos administradores de rede, uma grande quantidade de dados para tomada de decisões baseados no curso formal das ações em face da atividade suspeita. Este sistema também fornece uma solução rápida, através da criação de novas regras, para novos problemas na segurança de redes quando um novo ataque surge e os fabricantes demoram a lançar correções.

Os sistemas de detecção de intrusão comerciais mais atuais custam muito caro, enquanto o Snort está disponível gratuitamente e é livre para uso em qualquer ambiente compatível, tornando-se uma alternativa como sistema de segurança.

4.8. Características de um IDS peso leve

Um IDS de peso leve é composto pelas seguintes características (ROESH, 199):

- ?? Interferir o mínimo possível em outras operações do sistema;
- ?? Utilizar poucos recursos do sistema;
- ?? Facilitar a configuração pelos administradores do sistema;
- ?? Ocupar pouco espaço em disco;
- ?? Ser flexível bastante para ser utilizado como elemento permanente da infra-estrutura da rede;

4.9. Funcionamento do Snort

O Snort funciona habilitando o modo promíscuo da placa de rede do computador onde está instalado. O modo promíscuo de uma placa de rede, é aquele em que a placa de rede passa a receber todos os pacotes que estão trafegando por aquele segmento. Depois de habilitar a placa de rede em modo promíscuo, o Snort passa a capturar todos os pacotes para que sejam analisados através de um conjunto de regras, possibilitado assim, determinar se há um ataque ocorrendo. Através destas regras, que são conhecidas por assinaturas, é possível detectar uma grande quantidade de ataques e enviar alertas aos administradores em tempo real. A arquitetura do Snort, foi focada visando um melhor desempenho, simplicidade na criação de regras e flexibilidade.

O Snort é dividido em três módulos (ROESCH, 1999):

?? Decodificador de pacotes

?? O sistema de detecção

?? O sistema de *login* e alerta

As regras no Snort são simples e fáceis de escrever mas mesmo assim tem poder suficiente para detectar uma grande variedade de hostilidades ou tráfego de rede meramente suspeito. Existem três diretivas de ações básicas que podem ser usadas quando um pacote encaixa em um padrão de regra específico: *pass, log* ou *alert.* A diretiva *pass* simplesmente ignora o pacote deixando-o passar. A diretiva *log* escreve o pacote completo na rotina de registro selecionada pelo usuário em tempo de execução. A diretiva *alert*, gera um evento de notificação usando o método especificado pelo usuário na linha de comando e então registra o pacote completo usando mecanismo de registro de *log* selecionado para permitir uma análise posterior (ROESCH, 1999).

4.10. Regras do Snort

As regras mais básicas contém somente o cabeçalho, que é composto pelo protocolo, pela direção e a pela porta de interesse, como mostra a figura abaixo.

Uma regra básica do Snort

```
Log tcp any any -> 10.1.1.0/24 79
```

Figura 9 - Uma regra simples do Snort

Esta regra deve gravar (*log)* todo tráfego originado na porta 79 (*finger*) direcionado para a sub-rede classe C 10.1.1.

Incrementeando as regras mais básicas, é possível colocar opções na seqüência e entre parênteses. O Snort interpreta todas as palavras entre parênteses como campos de opções que estão disponíveis para todos os tipos de regras e podem ser usados para gerar comportamentos complexos de programas como na figura abaixo.

```
alert tcp any any -> 10.1.1.0/24 80 (content: "/cgi-bin/phf": msg:
"PHF probe!";)
```

Figura 10 -Regra do Snort incrementada com o opções

A primeira parte da regra contém a ação (*alert, log, pass, activate* ou *dynamic*), a segunda parte da regra, contém o protocolo a ser analisado, sendo que atualmente o Snort suporta três protocolos (TCP, UDP e ICMP) podendo no futuro suportar outros. O terceiro elemento, é o endereço IP e a porta ao qual o ataque é dirigido, a palavra-chave *any* pode ser utilizada para generalizar qualquer endereço IP. Não existe nenhum mecanismo para resolver a busca pelo nome do *host*, sendo assim necessário informar o endereço IP numérico e uma máscara para definição da classe. Em seguida, define-se o número da porta ou a palacra-chave *any* para qualquer porta, podendo ser definidos intervalos de portas utilizando o sinal ":" . Como por exemplo, o intervalo de portas entre 1 e 1024 pode ser escrito com 1:1024.

O operador de direção "->" indica a direção do tráfego de informações para que a regra seja aplicada. O endereço IP e porta do lado esquerdo do operador de direção informa a origem e do lado direito informa o destino. Pode ser usado o operador "<>" para indicar análise em ambas as direções.

A seção de opções combina facilidade de uso com poder e flexibilidade. Todas as opções de regras são separadas por um caracter ";", enquanto as palavras-chave são separadas pelo caracter ":", sendo atualmente definidas vinte e quatro palavras-chave para a escrita de regras:

- ack: Procura por um número específico no campo reconhecimento do cabeçalho TCP;
- content: procura por uma padrão específico na carga do pacote;
- content_list: procura por um conjunto de padrões na carga do pacote;
- depth: modificador da opção *content*. Muda *depth* para máximo de procura em um novo início de busca;
- dsize: testa o tamanho dos pacotes comparando com o tamanho especificado;
- flags: testa as *flags* TCP para valores especificados;
- fragbits: testa os bits fragmentados do cabeçalho IP;

?? icmp_id: testa o número seqüencial do campo ICMP ECHO ID contra um valor especificado;

?? icmp_seq: testa o número seqüencial do campo ICMP ECHO contra um valor especificado;

?? icode: testa o campo *code* do ICMP com um valor especificado;

?? id: Testa o cabeçalho IP por um valor específico;

?? ipoption: procura nos campos do protocolo IP por opções;

?? itype: testa o campo *type* do ICMP com um valor especificado;

?? logto: armazena a informação capturada em um arquivo especificado pelo usuário;

?? minfrag: ajusta o valor limiar para o tamanho de fragmento IP;

?? msg: registra uma mensagem no arquivo de alerta e captura o pacote;

?? nocase: habilita a procura por situações em pacotes independentes dos dados estarem representados em letra maiúscula ou minúscula;

?? offset: modificador da opção *content*. Ajusta o *offset* para uma nova busca;

?? react: respostas ativas (blocos de páginas web);

?? resp: respostas ativas (em caso de queda de conexão);

?? rpc: verifica serviços de RPC (*Remote Procedure Call*) para chamadas de aplicações/procedimentos específicas;

?? seq: Testa o campo “número seqüencial” do cabeçalho TCP para um valor especificado;

?? session: copia a informação do nível de aplicação para uma determinada sessão;

?? tos: testa os cabeçalhos IP com um valor específico para o campo TOS;

?? ttl: testa os cabeçalhos IP para um valor específico para o campo TTL;

5. REDES NEURAIS E SUA UTILIZAÇÃO EM IDS

5.1. Introdução às Redes Neurais

Um modelo de rede neural é identificado pela sua topologia e pelo seu método de aprendizado. Como as redes neurais possuem inspiração biológica, elas assemelham-se a modelos neurais do cérebro humano. O modelo artificial pioneiro de neurônio biológico foi proposto por McCulloch e Pitts em 1943. Após esta proposição, várias atualizações e várias idéias foram surgindo para adaptar as redes neurais aos problemas de cunho real.

A aprendizagem em redes neurais é caracterizada pela capacidade que as redes possuem de modificar o seu comportamento em resposta a eventos ou situações que ocorrem no ambiente externo e que fornecem um conjunto de entradas, o qual pode ser associado a um conjunto de saídas desejadas ou não. Através de um algoritmo de treinamento, este conjunto de entrada acarreta um ajuste dos pesos da rede, produzindo um conjunto de resposta adequado que concorda com os padrões de entrada ou como os padrões armazenados pela rede. Após a execução consistente e correta deste aprendizado, a rede torna-se capaz de compor similaridades e generalizar situações que ainda não foram aprendidas. Durante o treinamento da rede, é muito importante a monitoração de quanto tempo ela deve ficar treinando, pois um treinamento muito prolongado pode levá-la a um estado de especialização, nesta situação, a rede perde a capacidade de generalização, pois tende a decorar os padrões de entrada.

O comportamento de uma rede neural, depois de treinada, é determinado pelos pesos existentes entre as conexões de seus neurônios e as funções de ativação usadas para o treinamento da rede. Estas funções são consideradas os limiares de ativação da rede. Toda rede neural possui uma topologia que está ligada diretamente ao problema que se deseja resolver, à complexidade deste problema e a outras abordagens.

Entre as muitas áreas de aplicação de Redes Neurais, a principal é o reconhecimento de padrões. Do ponto de vista humano, o reconhecimento de um padrão, seja ele qual for, compreende a técnica pela qual uma pessoa, uma vez havendo aprendido a reconhecer determinado assunto, poderá reconhecê-lo outra vez, mesmo que o que ela esteja observando novamente não seja exatamente igual ao que lhe foi apresentado antes. O mesmo se aplica a uma rede neural, ou seja, uma vez que ela seja exposta a um conjunto de casos-padrão, passa a reconhecer situações semelhantes.

Um aspecto comum da implementação de redes neurais é considerar o cérebro como um dispositivo computacional paralelo, muito diferente dos computadores seriais tradicionais. McCulloh e Pitts (1943) propuseram uma unidade binária com limiar de ativação como um modelo computacional de um neurônio. O neurônio matemático calcula uma soma ponderada de *n* sinais de entrada *xj*, com *j = 1, 2, 3...,n,* e gera uma saída de *1* se a soma está acima de um certo limiar *u*. Esse limiar é denominado *threshold*.

O processo computacional envolvido com uma rede neural artificial (RNA) é desenvolvido da seguinte maneira: um neurônio artificial, ou elemento de processamento, recebe entradas de um grande número de outros neurônios artificiais, ou de uma fonte de estímulo externo (Figura 11).

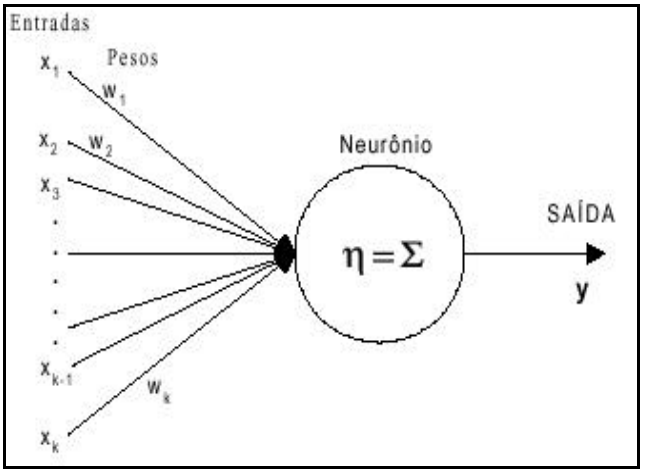

Figura 11 -Modelo de McCulloch-Pitts para um neurônio

Uma soma ponderada, chamada valor de ativação, dessas entradas constitui o argumento, para uma função de ativação (ou função de transferência). O valor de ativação ? *k* , para uma unidade de neurônio *k* num tempo *t* é dado por:

$$? _{k(t)} ?? ?? _{j} w_{kj} x_j(t) + ? _k$$

onde , w_{kj} é o peso na conexão (sinapse) do j-ésimo neurônio, x_j é o valor de saída do neurônio *j* e *?k* é um valor de ajuste que está relacionado com o limiar de ativação.

O modelo neuronal de McCulloch-Pitts tem sido generalizado de muitas maneiras. Uma delas é a utilização de uma função de ativação que não seja necessariamente a função degrau. A função de ativação, que define as propriedades do neurônio, é geralmente não-linear. O valor resultante da função de ativação é a saída do Neurônio Artificial. A magnitude da saída, e o peso das conexões, determinam o efeito da unidade.

Uma função não-linear típica, mais usada em redes neurais, é a função sigmoidal logística, dada por:

$$y_k(t) = \frac{1}{1 + e^{\frac{-\eta_k(t)}{T}}}$$

onde *T* é um parâmetro de ajuste da declividade da função. Esta saída se distribui a outros Neurônios Artificiais ao longo de conexões ponderadas, formando a topologia. A arquitetura da rede é a maneira na qual estas conexões são feitas, o que define o fluxo de informação. As Redes Neurais Artificiais podem ser vistas como grafos ponderados, nos quais os Neurônios Artificiais são os nós, e as setas direcionais (com os pesos) são conexões entre saídas e entradas dos neurônios. Baseado na arquitetura, ou seja, no modelo de conexão, as Redes Neurais Artificiais podem ser divididas em duas categorias (JAIN et al.,1996):

?? Redes diretas (“*feedforward*”), nas quais os grafos não possuem retorno ou ciclos (“*loopings*”).

?? Redes com ciclos ou recorrentes (“*feedback*”), nas quais os ciclos (“*loopings”)* ocorrem devido a conexões de realimentação.

Na família de redes diretas mais comum, chamada *Multilayer Perceptron*, os neurônios são organizados em camadas que possuem conexões unidirecionais entre eles. Conectividades diferentes resultam em diferentes comportamentos para a rede. Genericamente falando, redes diretas são estáticas, isto é, elas produzem, a partir de uma dada entrada, somente um conjunto de valores de saída, ao invés de uma seqüência de valores. Redes diretas são ditas sem-memória, no sentido de que sua resposta para uma dada entrada é independente do estado anterior da rede. Por outro lado, redes recorrentes, ou tipo *feedback*, são sistemas dinâmicos. Quando um novo padrão de entrada é apresentado, as saídas do neurônio são calculadas e, devido aos desvios de realimentação, as entradas de cada neurônio são então modificadas, o que leva a rede a entrar num novo estado.

As conexões ponderadas das arquiteturas possuem um papel muito importante nas redes neurais. O método usado para ajustar os pesos no processo de treinamento de uma rede é chamado de regra de aprendizagem. Diferentes arquiteturas requerem algoritmos de aprendizagem apropriados. A aprendizagem pode ser supervisionada ou não-supervisionada. A regra de aprendizagem supervisionada, mais amplamente utilizada, é o método de *backpropagation*. Outro tipo de aprendizagem, entretanto não-supervisionada, é o método conhecido como auto-organização.

Em resumo, os três componentes essenciais de um sistema computacional baseado em redes neurais artificiais são: a função de ativação, a arquitetura e a regra de treinamento. Um dispositivo computacional para implementar simulações de cadeias neurais artificiais, de um modo parecido com o cérebro humano, consiste de diversas unidades neurais citadas acima, ricamente conectadas umas às outras.

5.2. A aprendizagem da rede neural

Não basta conectar neurônios para que eles forneçam um resultado útil. É necessário um método para treiná-los. Esse mecanismo deve ser tão simples quanto possível, de forma que sua modelagem não se torne extremamente complexa. O treinamento consiste basicamente em reforçar bons comportamentos, que devem ser repetidos, e reprimir os maus. O princípio chave no processo de treinamento de uma rede neural é deixá-la aprender com seus próprios erros. Se ela produz uma saída

incorreta, deve-se reduzir as chances desta se repetir, e caso produza um valor desejado, nenhuma medida é tomada.

O processo de treinamento, ou aprendizagem, de uma rede neural está relacionado com a atualização da arquitetura da rede e dos pesos das conexões, de forma que a rede possa realizar eficientemente uma determinada tarefa. A rede usualmente deve aprender os pesos das conexões, por intermédio de modelos de treinamento. As RNAs possuem a capacidade de aprender a partir de exemplos, ou seja, ao invés de seguir um conjunto de regras estabelecidas por especialistas humanos, elas podem aprender a partir de uma dada coleção de exemplos representativos. Esta é uma das maiores vantagens das redes neurais sobre os sistemas especialistas tradicionais.

Existem três paradigmas principais de aprendizagem (HAYKIN, 1994): supervisionado, não supervisionado e híbrido. Na aprendizagem supervisionada é fornecida à rede a resposta correta para cada modelo de entrada. Os pesos são determinados de maneira que a rede produza respostas o mais próximo possível das respostas corretas conhecidas. Há uma variante da regra supervisionada, conhecida como aprendizagem reforçada, na qual são fornecidas à rede apenas as críticas (correções) às respostas de saída, e não as respostas corretas em si. Por outro lado, aprendizagem não supervisionada não requer uma resposta correta associada com cada modelo de entrada no conjunto de dados de treinamento. Ela explora a estrutura adjacente, ou correlações entre padrões dentro do conjunto de dados, e os organiza dentro de categorias, a partir daquelas correlações. Já o treinamento híbrido combina aprendizagem supervisionada e não supervisionada, com partes dos pesos sendo determinados através de cada método.

Os paradigmas são tratados a partir de teorias de aprendizagem, que resultam em regras para os algoritmos apropriados. Estes algoritmos, por vezes, possuem alta complexidade computacional. Existem quatro tipos básicos e regras de aprendizagem: regra de correção de erro, regra de Boltzmann, regra de Hebbian e regra de treinamento competitivo. Ambos os paradigmas, supervisionados e não supervisionados, empregam regras baseadas em correção de erros, regra de Hebbian e aprendizagem competitiva. Regras baseadas em correção de erro podem ser usadas para treinar redes diretas, enquanto regras de aprendizagem de Hebbian, tem sido usadas para todas as

arquiteturas de rede. Entretanto, cada algoritimo de aprendizagem é projetado para treinar uma arquitetura específica. Desta forma, um algoritmo de aprendizagem, está associado à uma arquitetura particular.

5.3. O re-treinamento e adaptabilidade da rede

A capacidade de re-treinar uma rede neural fornece características muito interessantes. Sempre que se queira introduzir um novo padrão para que ela passe a reconhecê-lo, basta re-treiná-la, só que agora com este novo padrão fazendo parte do conjunto de padrões de treinamento. Esta característica fornece uma habilidade de adaptação que é necessária em algumas aplicações, notadamente quando o conjunto de padrões que a rede terá de reconhecer não é constante, mas pode apresentar variações no decorrer do tempo. Num sistema de reconhecimento de caracteres esta é uma propriedade não tão útil, já que é extremamente difícil ocorrer uma mudança nas letras que formam o alfabeto. Já num sistema de detecção de intrusão, esta característica é de fundamental importância, pois confere ao sistema este poder de adaptabilidade.

5.4. Modelos De IDS Baseados em Redes Neurais

As redes neurais são conhecidas pela sua alta capacidade de adaptação, aprendizado e generalização. Os modelos baseados em redes neurais visam explorar estas características e, através de treinamento, gerar uma estrutura que seja capaz de classificar padrões de intrusão ou normalidade.

Estes modelos podem ser usados para definir tanto sistemas de detecção por anomalia quanto sistemas de detecção por abusos, bem como sistemas baseados em servidores ou baseados em rede. Para que seja possível desenvolver um sistema destes, é necessário definir qual será a topologia da rede, seu algoritmo de treinamento, as variáveis quantitativas e qualitativas que representem o modelo e também os dados que irão compor o treinamento da rede. Além disso, deve-se definir como serão feitas as verificações e adaptabilidades dos dados após a rede ter sido treinada.

Sistemas de detecção de intrusos por anomalia usando redes neurais podem ser vistos em: Debar (DEBAR et al., 1992), que propôs um sistema online que aprende a predizer qual será o próximo comando a ser usado por um usuário do sistema; Ryan

(1998) que propôs um sistema offline usando métricas referentes à freqüência de utilização de comandos para identificar o legítimo perfil de um usuário e Tan (1995), que usou várias métricas para compor um vetor de dados a ser aplicado à rede neural, objetivando assim a detecção de padrões inesperados em sessões de uso destas métricas.

Um sistema de detecção de intrusos por abuso usando redes neurais foi proposto por Cansian (1997) onde um módulo usando uma rede neural do tipo Multi-Layer Perceptron com 126 entradas e 1 saída foi utilizado para testar assinaturas de intrusão previamente treinadas e verificar se o sistema está ou não sob ataque.

Outro ponto importante na utilização de modelos baseados em redes neurais para detecção de intrusão por anomalia é a forma como será tratada a entrada de dados. Em (RYAN et al., 1998) uma discussão é feita a respeito da determinação da janela a ser aplicada a uma rede de detecção de comportamento intruso. Esta janela indica quantos comandos serão capturados para serem aplicados a entrada da rede neural. A rede neural é recorrente, isto é, parte da saída é retornada para a entrada do próximo passo. Assim, ela está sujeita a "esquecer" os padrões mais antigos de comportamento com o passar do tempo. Desta forma se o tamanho da janela for muito pequeno, ocorrerão muitos falso-positivos; e se o tamanho da janela for muito grande, a rede não generalizará bem novos perfis, isto é, poderão ocorrer mais falsos-negativos.

5.5. NNID - Neural Network Intrusion Detection

O Sistema de Detecção de Intrusão com Rede Neural proposto por Jake Ryan (RYAN et al., 1998), é baseado na identificação de intrusão por anomalia, e procura identificar um usuário legítimo baseando-se na distribuição de comandos usados por eles. Isto é possível, pois diferentes usuários tendem a exibir comportamentos diferentes dependendo das necessidades de suas tarefas no sistema. Alguns usuários enviam emails e utilizam editores de texto, outros usuários utilizam planilhas de cálculos e outros usuários utilizam aplicações específicas. Ainda que dois usuários realizem as mesmas tarefas, pode ser que prefiram aplicativos diferentes. A freqüência que alguns comandos são utilizados também costuma variar de usuário para usuário. O conjunto de comandos utilizados e sua freqüência, constitui uma marca de seu uso refletindo a tarefa executada e a escolha do programa aplicativo, sendo possível assim identificar o usuário baseado

nestas informações. A privacidade do usuário será mantida uma vez que os argumentos utilizados nos comandos não serão registrados, isto é, poderá ser gravado um *log* identificando que um usuário envia cinco email por dia, mas não será necessário identificar para quem ele enviou.

Ryan construiu um NNID para um sistema de rede específico da utilizando as três fases seguintes:

1. Colhendo dados para o treinamento da rede neural: Obteve registros de auditoria de usuários por um período de muitos dias. Para cada dia e usuário, formou um vetor que representa o quanto um usuário executou cada comando.
2. Treinando a rede: Treinou a rede neural para identificar o usuário baseado nos vetores de distribuição de comandos.
3. Execução: Deixou a rede identificar um usuário para cada novo vetor de distribuição de comandos. Se a sugestão da rede for diferente do usuário atual ou se a rede não tiver uma sugestão clara, é sinal de anomalia.

O experimento foi executado em um servidor de um laboratório com dez usuários, todos conhecidos, alguns regulares e outros eventuais, foram analisados um total de 100 comandos, os mais comuns registrados nos *logs*, para descrever o comportamento destes usuários. Esta configuração foi escolhida por ser de um tamanho que seja possível gerenciá-la e manter a sua confiabilidade e poder ser comparada com um sistema do mundo real.

A rede neural utilizada foi montada com três camadas utilizando a arquitetura *multilayer-perceptron*. A camada de entrada formada por 100 unidades representadndo o vetor de usuário, a camada escondida é composta por 30 unidades e a camada de saída por dez unidades, uma para cada usuário. A rede foi treinada utilizando 8 dias aleatórios totalizando 65 vetores, e a execução foi testada utilizando os 4 dias restantes utilizando um total de 24 vetores.

5.6. Um Modelo Adaptativo de Detecção de Intrusos

O modelo proposto por Adriano Mauro Cansian (1997), trata de um sistema de segurança para redes de computadores que operem utilizando protocolo TCP/IP. O

comportamento intrusivo é detectado, através da localização de assinaturas de ataque no fluxo de dados da rede. Trata-se de um modelo de detecção por abusos, que utiliza técnicas de captura de pacotes, aliadas a uma rede neural, para além de descobrir o comportamento intrusivo, auditar e fornecer elementos que auxiliem no processo de decisão sobre possíveis ações a serem tomadas pelo administrador. O sistema de detecção é composto por um ou mais sistemas de captura de pacotes que, posicionados em locais estratégicos da rede, obtém as informações que são tratadas por um sistema de segurança.

Existem algumas técnicas de acesso, pelas quais as tentativas de ataque se desenvolvem. Na grande maioria dos cenários, o intruso se encontra fisicamente distante do sistema sob ataque (NEUMANN, 1989), e assim utiliza algum ponto de rede durante sua ação. A Figura 12 ilustra quatro tipos de comportamentos básicos de ataque (KO et al., 1993).

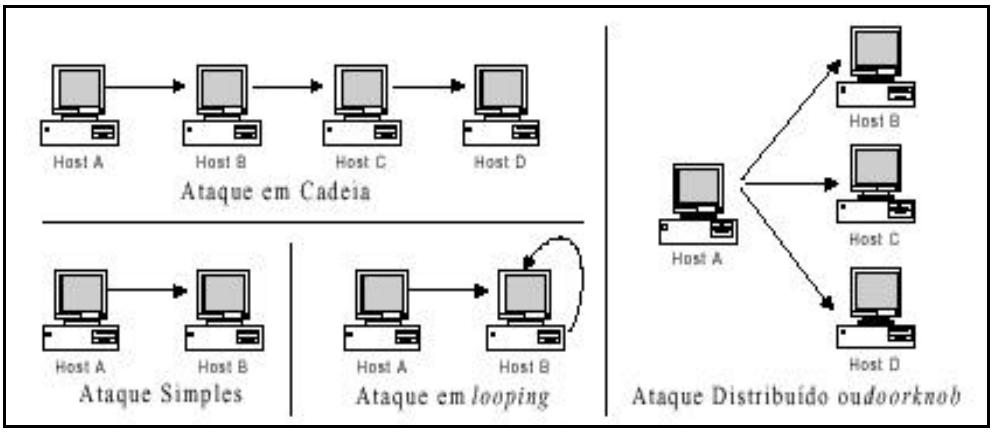

Figura 12 - Tipos básicos de ataque

No ataque simples o intruso usa uma máquina na rede para acessar diretamente outra máquina. No ataque distribuído (também conhecido como *doorknob*), diversas tentativas de acesso (com ou sem sucesso) se desenvolvem a partir de uma única máquina. No ataque em cadeia, o intruso procura acessar diversas máquinas para dissimular sua verdadeira identidade, ou seu ponto de ataque inicial; no ataque em *looping*, a máquina atacada é a mesma originadora do ataque.

Uma das propostas do sistema, é a introdução de um mecanismo de segurança, capaz de detectar comportamento intrusivo em determinadas sessões de rede. Este mecanismo atua capturando e decifrando pacotes, que são transmitidos através da rede

sob monitoração. Para realizar inferências acerca do estado de segurança das sessões, utiliza um sistema de pré-filtragem e uma rede neural. Como resultado, a rede neural fornece um número que, baseado em informações de assinaturas de ataques anteriores, indicará a gravidade de um determinado evento.

O sistema se baseia no fato de que um comportamento intrusivo pode ser detectado a partir da análise de padrões pré-determinados.

Estes comportamentos, desde que devidamente observados e tratados, geram assinaturas de ataque que identificam um determinado padrão, que por sua vez pode ser identificado nos dados das sessões de rede. A proposta apresentada é que a utilização de redes neurais fornece mecanismos eficazes de reconhecimento de ataque, além de inserir uma capacidade adaptativa que acompanhe as mudanças e variações nas técnicas existentes.

O sistema chamado de agente é implantado em uma máquina segura, ou seja, um computador logicamente invisível às outras máquinas, e que se encontra num local onde o acesso físico é restrito, podendo assim ser acessado apenas em determinadas condições especiais. Também podem ser utilizadas técnicas especiais de ocultação, utilizando, por exemplo, uma alocação dinâmica de números IP, ou ainda um conjunto de máquinas. Este sistema seguro é posicionado em pontos sensíveis do sistema de rede.

5.6.1. Estrutura do Modelo

O Modelo Adaptativo de Detecção de Intrusos é formado por um sistema de captura e tratamento de pacotes, um sistema de rede neural, e um gerenciador de comunicações e interface com o usuário (Figura 13).

O sistema de captura e tratamento de pacotes é organizado em módulos, que tratam o fluxo de pacotes, e que terminam fornecendo o vetor de estímulo para a rede neural.

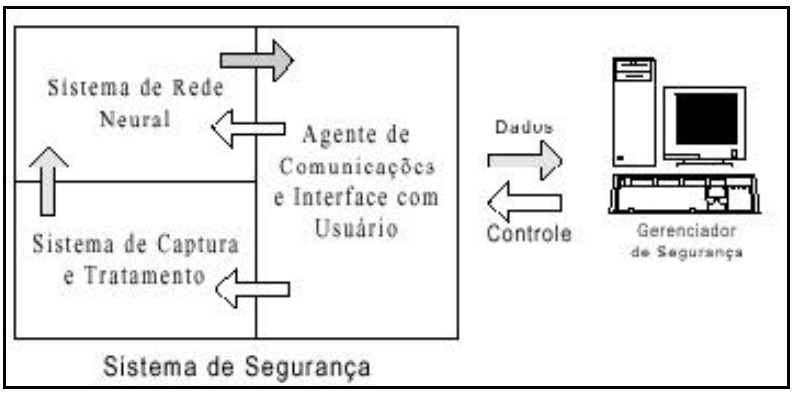

Figura 13 - Estrutura do Modelo Adaptativo de Detecção de Intrusos

5.6.2. Obtenção do Nível de Supeita da Sessão

A rede neural analisa o vetor de estímulo de cada sessão, com os respectivos pesos que representam a importância da ocorrência dos eventos, e procura atribuir um grau de suspeita, que representa o estado de segurança da sessão. Antes que a rede neural possa identificar potenciais ataques, esta necessita ser treinada com um conjunto significativo e suficientemente grande de vetores de estímulo, que representem tanto o comportamento de sessões suspeitas e intrusivas quanto legítimas. Uma vez treinada, a rede deve utilizar sua propriedade de generalização para identificar corretamente usuários que apresentem características semelhantes àquelas presentes nas ações de intrusão utilizadas para treiná-la.

O sistema de gerenciamento (remoto ou não) recebe então o valor numérico que representa o estado de segurança das sessões suspeitas e as mapeia, classificando-as pelo grau de severidade, por intermédio de um código de cores. A partir deste ponto, várias ações podem ser tomadas, tais como emissão de vários níveis de alertas aos administradores (dependendo do nível de risco dos eventos detectados), disparo de processos de auditoria, ativação de contra-medidas para isolar o *host* ou o domínio que originou o ataque, através do ajuste ativo de *wrappers* (VENEMA, 1992), *firewalls* ou filtros. O sistema pode ainda sugerir ao administrador qual a melhor medida a ser tomada para sanar o problema, ou ainda, tentar localizar e contactar o administrador responsável pelo sistema.

6. TRABALHO REALIZADO

Os estudos na área de segurança, especificamente a detecção de intrusos em redes de computadores mostram que a maioria das técnicas empregadas atualmente consegue detectar muitos ataques já conhecidos, mas tem dificuldades quando se trata de um ataque desconhecido, ou de uma variação dos ataques já codificados.

Este trabalho foi realizado com a intenção de desenvolver técnicas de detecção de intrusos baseado em servidor em uma rede de acesso público utilizando redes neurais para verificar a identidade dos usuários baseado em seu comportamento.

O desenvolvimento de uma rede neural para avaliar os registros de auditoria dos IDSs atuais pode fazer com que a detecção seja mais segura, eficaz e com um bom desempenho diferenciando-se assim dos sistemas de detecção de intrusão baseado em regras que detectam apenas ataques conhecidos.

O ambiente escolhido para os testes foi o Laboratório de Informática do Curso de Ciência da Computação da Universidade de Passo Fundo - LabComp. O LabComp, conta com dois servidores Linux Conectiva[6] versão 6 servindo os computadores dos funcionários e professores do curso e mais 20 computadores para utilização dos alunos no desenvolvimento de suas tarefas acadêmicas. O primeiro servidor é destinado aos recursos de hardware e software servidos aos alunos do curso, enquanto o segundo serve aos funcionários e professores. Neste laboratório, foram selecionadas vinte contas de usuários a serem monitoradas através de um módulo instalado no segundo servidor, ou seja, com o intuito de registrar o padrão de uso destes usuários em um IDS baseado em servidor para detectar ataques por anomalia.

[6] Distribuição do Sistema Operacional Linux disponível em http://www.conectiva.com.br

6.1. Redes Neurais para IDS Baseado em Servidor

Um dos problemas mais comuns em laboratórios para alunos de um curso de informática é a utilização indevida dos recursos autorizados a uma conta de usuário por outro usuário que consegue, de formas diversas, se fazer passar pelo usuário autorizado.

Para conseguir detectar alguns abusos, tem-se a alternativa de utilizar um Sistema de Detecção de Intrusos por Anomalia, monitorando as atividades dos usuários da rede. Baseado nos trabalhos de Ricardo Bernardo dos Santos(2000) e Jake Ryan (1998), foi desenvolvido um módulo para capturar os registros de auditoria de um servidor de autenticação baseado em Linux, transformá-los em assinaturas para alimentar uma rede neural *Multilayer Perceptron* que identificará se o padrão capturado está dentro dos padrões normais daquele usuário.

O trabalho foi dividido em três etapas:

1. A coleta de dados para o treinamento da rede neural;

   Nesta etapa é feita a leitura de um arquivo de registro de auditoria (.bash_history) de comandos executados por cada usuário, realizando uma contagem dos comandos mais executados para a geração de um vetor de distribuição de comandos diário individual para cada usuário analisado (conforme descrito na seção 6.2) representando a freqüência com que cada comando é executado individualmente;

2. O treinamento da rede neural;

   Após um período de cinco dias de coleta, análise e preparação dos vetores de distribuição de comandos, estes foram utilizados para treinar a rede neural para que esta pudesse registrar e 'aprender' o padrão de comportamento que cada um dos usuários apresenta em suas seções de trabalho;

3. A execução dos testes da rede neural com dados diários não utilizados em treinamento;

   Coma a rede neural treinada, iniciou-se a fase de testes com a alimentação de novos vetores de distribuição de comandos coletados dos mesmos usuários em mais cinco dias diferentes. Ao receber um vetor de comandos

de um usuário em seus neurônios de entrada, processar este padrão e responder com um valor acima de um limiar definido em um dos neurônios de saída, este indicará o usuário. Se resposta da rede for diferente do usuário apresentado, ou se todos os neurônios de saída da rede estiverem abaixo do limiar, a resposta é um sinal de uso anormal, o que pode ser um indicativo de intrusão.

A escolha dos dez usuários a serem analisados foi feita entre os professores e funcionários do curso que utilizam com freqüência o LabComp para tentar garantir uma certa confiabilidade nos resultados obtidos, pois são usuários regulares e conhecidos. A partir do aceite destes usuários em participar desta pesquisa, passou-se a registrar seus padrões de comportamento por cinco dias consecutivos no uso de comandos em suas sessões de acesso ao servidor utilizando terminal remoto em sessões SSH, para formar os vetores de comandos que foram usados na fase de treinamento e mais cinco dias para formar os vetores de comandos que foram utilizados nos testes da rede neural.

Existia a preocupação de respeitar o sigilo das operações realizadas por estes usuários, o que foi garantido, pois como são coletados apenas os comandos que são executados e não há a necessidade de registrar os parâmetros que acompanham cada comando, com isso sabemos, por exemplo, que um determinado usuário faz, em média, cinco cópias de arquivos por dia, mas não sabemos quais são estes arquivos e nem para onde são copiados, um outro usuário envia em média seis *e-mails* por dia, mas não é necessário saber para quem ele envia e tampouco o conteúdo destes *e-mails*.

6.2. Vetor de Distribuição de Comandos

O sistema operacional Linux gera um arquivo chamado *".bash_history"* (Figura 14) que registra todos os comandos executados por cada usuário e que fica armazenado na pasta pessoal do usuário. Este arquivo acumula todos os comandos executados pelo usuário em todas as sessões de trabalho, isto é, não possui um indicativo de quando os comandos foram executados, ou de quantos comandos foram executados em uma sessão de trabalho. Isto dificultaria o trabalho de coleta de informações para gerar um perfil diário de utilização de comandos por um usuário.

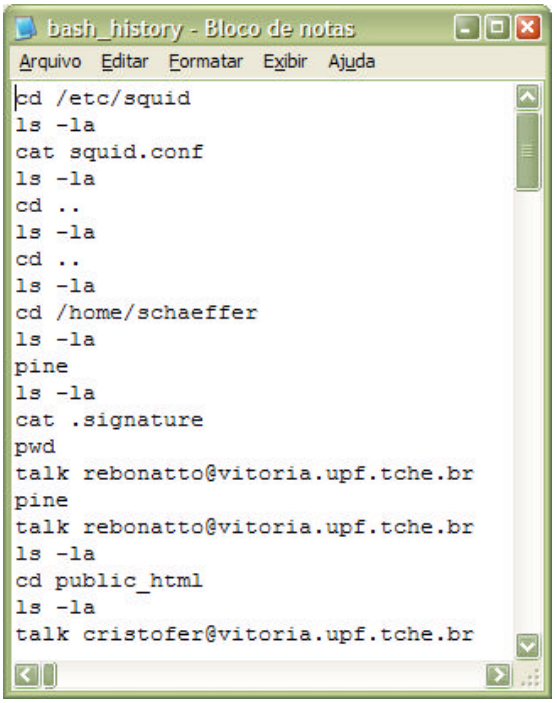

Figura 14 - Arquivo de registro de comandos executados original do Linux

Para resolver este problema, foi desenvolvido um sistema que gera um outro arquivo (Figura 15) que ficará armazenado na pasta "*home*" do usuário "*root*" identificando-o com o nome de usuário (*login*) e a data de execução e que registra apenas os comandos executados, sem registrar seus parâmetros, e marca com separações indicando a data de início e de final de cada sessão de trabalho.

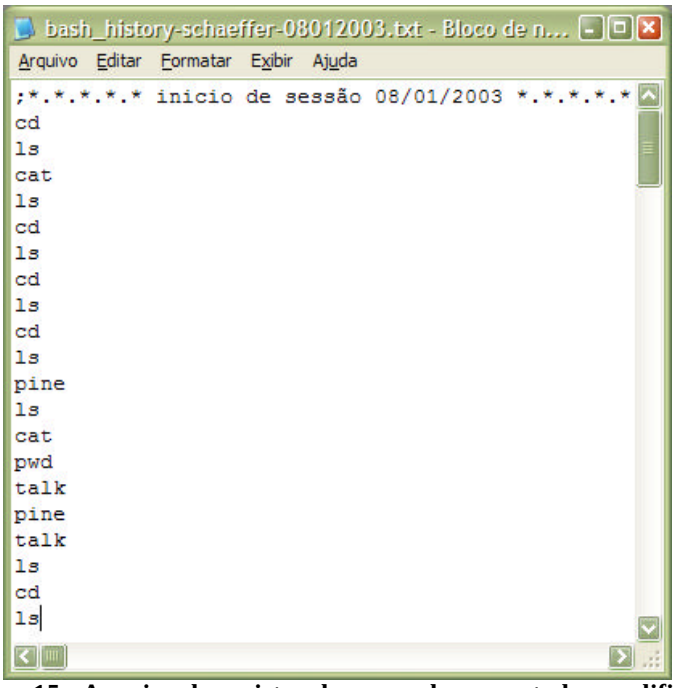

Figura 15 - Arquivo de registro de comandos executados modificado

Após a verificação dos registros coletados nos primeiros cinco dias de teste, verificou-se a presença de muitos comandos, sendo que vários comandos foram utilizados apenas uma vez em apenas uma sessão por um usuário. Entre todos os comandos registrados, foram selecionados apenas os cinqüenta comandos mais utilizados em todas as sessões (apresentados na Tabela 3). Esta tabela foi criada também para que o sistema ignore os comandos digitados de forma errada, como uma letra invertida ou uma letra errada. Como por exemplo, o comando *finger* que tenha sido digitado “finegr” ou “figner”.

Tabela 3 - Comandos usados para descrever o comportamento do usuário

cat	cd	chmod	cp	clear
date	filter	find	finger	ftp
gcc	gmake	grep	gzip	ifconfig
l	la	linuxconf	look	lpq
lpr	ls	make	man	mkdir
more	mv	netstat	passwd	perl
pgp	pico	pine	ping	os
pwd	quota	rename	rm	rmdir
ssh	sort	tail	talk	tar
telnet	vi	whereis	who	whois

A arquitetura da rede neural criada possui um neurônio de entrada para cada um destes cinqüenta comandos. A freqüência de utilização dos comandos por cada usuário dará origem ao perfil de comportamento deste usuário.

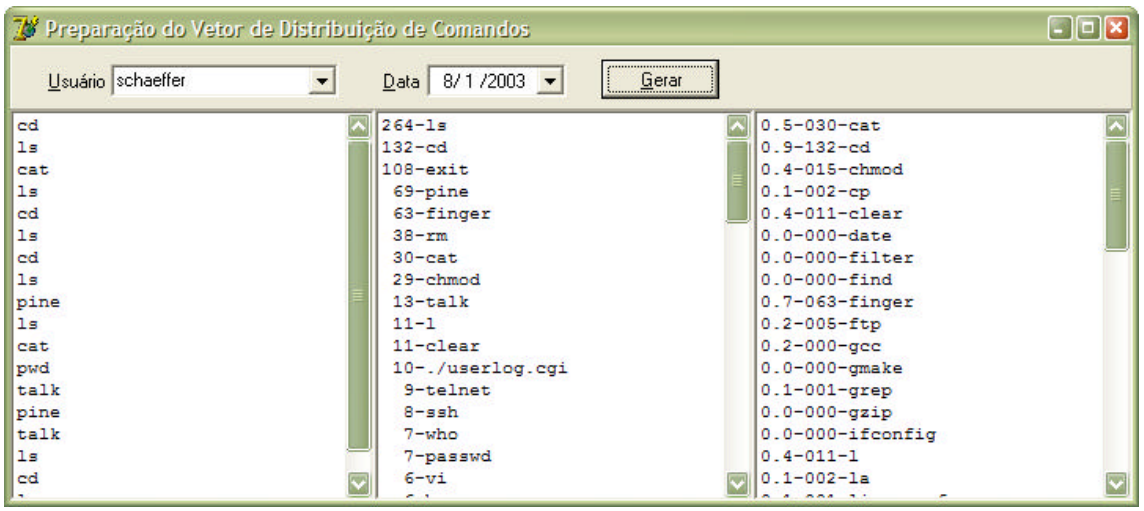

Figura 16 - Geração do Vetor de Distribuição de Comandos

Para efetuar a análise destes arquivos e a preparação do vetor de comandos, foi desenvolvido um programa que recebe uma data e o nome de *"login"* de um usuário e através destes dados, o programa encontra o arquivo de registros correto, faz a leitura dos comandos registrados totalizando a freqüência com que cada comando foi executado e preparando o vetor de comandos(Tabela 4) que irá alimentar a rede neural.

Tabela 4 - Vetor de distribuição de comandos

Qtd	Comandos	Valor
264	ls	1.0
132	cd	0.9
63	finger	0.7
38	rm	0.6
30	cat	0.5
29	pine	0.5
15	chmod	0.4
13	talk	0.4
12	ssh	0.4
11	clear	0.4
11	l	0.4
9	telnet	0.3
7	who	0.3
7	passwd	0.3
6	vi	0.3
5	man	0.2
5	ftp	0.2
4	pwd	0.2
3	rmdir	0.2
3	gcc	0.2
2	quota	0.1
2	la	0.1
2	tail	0.1
2	cp	0.1
1	grep	0.1

A análise de comportamento será feita pela rede neural através de um vetor criado a partir da contagem dos comandos digitados pelo usuário diariamente.

6.3. Arquitetura da rede neural

Foi desenvolvido um protótipo de rede neural baseado em uma arquitetura *multilayer-perceptron* com três camadas mostrada na Figura 17. A camada de entrada possui cinqüenta unidades, cada uma representando um dos comandos presente no vetor de distribuição de comandos do usuário; uma camada intermediária com 30 neurônios definida aleatoriamente por experimentação; e a camada de saída composta por 10

neurônios representando cada um dos dez usuários selecionados. Conforme Ryan (1998), muitas arquiteturas de redes neurais mais sofisticadas poderiam ser utilizadas, porém a idéia é obter os resultados de uma arquitetura mais genérica e simples para que a praticidade da pesquisa pudesse ser demonstrada e os resultados pudessem ser facilmente replicados. A natureza deste problema é estática, logo uma rede direta é mais adequada para este caso.

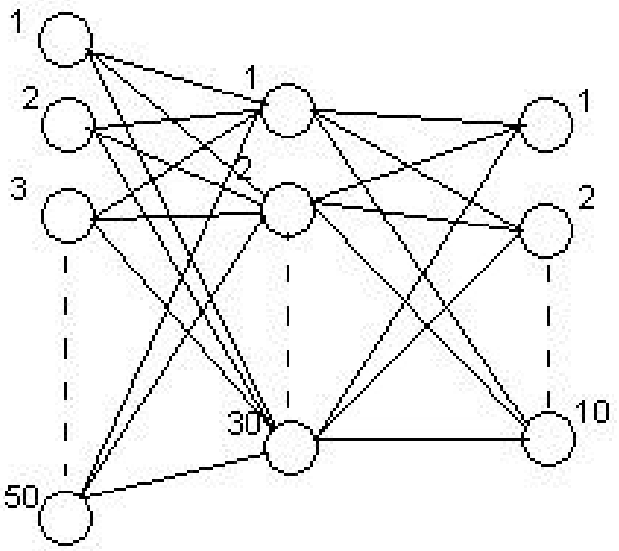

Figura 17 - Arquitetura da Rede Neural utilizada

6.4. Preparação dos vetores para treinamento

Durante cinco dias, no período de férias dos alunos, portanto, apenas alguns professores faziam uso do Laboratório, o que supostamente garante um pouco mais de confiabilidade nos dados registrados, foram coletados registros de auditoria com os comandos executados em sessões SSH de dez professores. Após, estes registros foram filtrados e apenas os cinqüenta comandos mais utilizados foram utilizados para a criação dos vetores diários de entrada para a rede neural.

De acordo com Ryan(1998), para conseguir uma maior integração entre os vetores de entrada e, portanto uma melhor generalização, o número de vezes que um comando foi utilizado foi dividido em intervalos (ver Tabela 5). Foram criados onze intervalos não linearmente espaçados para que a representação seja mais precisa nas menores freqüências que são mais importantes. O primeiro intervalo é para os comandos que nunca foram usados, o segundo intervalo para os comandos que foram usados uma ou

duas vezes, e assim por diante até o último intervalo onde os comandos foram utilizados mais de 200 vezes. Os intervalos estão representados por valores entre 0.0 e 1.0 com incremento de 0.1. Estes valores, um para cada comando, foram então concatenados em um vetor de distribuição de comandos com 50 posições para ser usado como entrada da rede neural.

Tabela 5 - Intervalos de execuções para gerar os valores de entrada na Rede Neural

Valores	Intervalos
0.0	comandos que nunca foram utilizados
0.1	comandos que foram utilizados de 1 até 2 vezes
0.2	comandos que foram utilizados de 3 até 5 vezes
0.3	comandos que foram utilizados de 6 até 10 vezes
0.4	comandos que foram utilizados de 11 até 20 vezes
0.5	comandos que foram utilizados de 21 até 30 vezes
0.6	comandos que foram utilizados de 31 até 50 vezes
0.7	comandos que foram utilizados de 51 até 70 vezes
0.8	comandos que foram utilizados de 71 até 99 vezes
0.9	comandos que foram utilizados de 100 até 199 vezes
1.0	comandos que foram utilizados de 200 ou mais vezes

Assim como no experimento de Ryan, para evitar o supertreinamento da rede, foram feitas várias sessões de treinamento para verificar quantos ciclos de treinamento resultavam em um melhor desempenho. A rede foi treinada com os vetores de comandos recolhidos dos cinco primeiros dias de teste.

6.5. Identificação dos usuários

A identificação do usuário é feita através dos valores apresentados pela rede neural nas unidades de saída. A unidade de saída que apresentar o maior valor e também for maior que um valor limiar, estabelecido em 0.5, será a indicação da rede neural para o usuário identificado. Se o usuário indicado pela rede neural for diferente do usuário que gerou o vetor, então estará identificado uma anomalia ou invasão.

Foram feitas várias sessões de treinamento, até definir a mais correta para um limiar de 0.5 para identificação de uma anomalia, isto é se a saída da rede apresentar um valor menor que 0.5, está caracterizada uma anomalia. Se nenhuma indicação da rede apresentar um valor maior que o limiar, então estará identificado um falso-positivo.

6.6. Resultados obtidos

Para a realização dos testes da rede neural, foram preparados cinqüenta vetores de comandos, coletados da utilização diária dos dez usuários que participaram na preparação dos vetores para a fase de treinamento. Para cada dia, foi preparado um vetor de comandos por usuário, totalizando ao final de cinco dias, cinqüenta vetores.

Estes vetores de comandos foram utilizados para testar a rede neural na identificação positiva dos usuários e esta apresentou 46 identificações corretas (92%) e quatro saídas (8%) identificando usuários diferentes dos apresentados, indicando uso anormal, ou seja, falso-positivos (ver Tabela 6).

Foram testados também (ver Tabela 7), vinte vetores, obtidos de sessões onde usuários foram convidados a realizar atividades em sessões de SSH de outros usuários. Neste caso, todos os vetores deveriam apontar usuários diferentes ou anomalia, mas dois vetores (10%) apresentaram o próprio dono da sessão de uso, ou seja, falso-negativos.

Tabela 6 - Vetores corretos apresentados para teste

Vetores apresentados para teste	Identificações corretas		Identificações negativas	
50	46	92%	4	8%

Tabela 7 - Vetores de intrusos apresentados para teste

Vetores apresentados	Identificações positivas		Identificações negativas	
20	2	10%	18	90%

Os trabalhos indicam um bom desempenho para a quantidade de usuários apresentada nos testes, e provavelmente devem manter estes padrões para uma quantidade maior de comandos selecionados para a composição dos vetores e também para um número maior de usuários.

7. CONCLUSÃO

Neste trabalho foram apresentados, conceitos importantes na área de segurança de informações, foi feita uma apresentação sobre algumas técnicas comuns de invasão em sistemas de rede e as principais técnicas utilizadas na defesa de sistemas de rede de computadores, seus recursos de hardware e software e suas informações.

Ficou bastante claro que os problemas de segurança afetam, nos dias atuais, todo tipo de atividades em todo o mundo, necessitando assim de políticas e atitudes específicas para tratar deste assunto. O avanço das pesquisas de novas técnicas para a solução destes problemas deve aumentar ainda mais, pois a criatividade dos atacantes certamente vai gerar novas técnicas de invasão.

Foi apresentado um estudo aprofundado sobre uma das técnicas utilizadas para a proteção dos sistemas de rede e suas informações, a Detecção de Intrusão. A classificação dos Sistemas de Detecção de Intrusão, suas características e potencialidades, seus benefícios também foram bastante estudados. Através deste estudo, nota-se a importância desta técnica para a segurança das informações e recursos de uma rede de computadores. Porém, ficou bastante visível que existe um problema bastante sério com os principais sistemas de detecção de intrusos, que é a detecção de novas técnicas de intrusão. Os administradores de rede ficam, quase sempre, na dependência dos fabricantes destes IDS fornecerem atualizações de seus produtos para os ataques que vão surgindo. Fica assim provado a necessidade de buscar métodos para auxiliar as análises feitas por estes sistemas de detecção, buscando encontrar variações de ataques conhecidos ou mesmo novos ataques.

A análise de comportamento de usuários, tentando descobrir uma mudança brusca em suas atividades diárias, é uma alternativa válida para a análise de assinaturas de ataque baseada em tráfego de rede.

A utilização de redes neurais no auxílio dos sistemas de detecção de intrusão por anomalia, apresentada neste estudo, provou ser capaz de bons resultados na identificação do usuário com forte tendência de evolução perante os métodos tradicionais de avaliação de intrusão. A descoberta de intrusos pode ser feita. As avaliações de comportamento obtidas através da análise de comandos executados em um ambiente real, por usuários regulares desenvolvendo trabalhos reais do dia-a-dia, garantem a confiabilidade dos testes efetuados neste trabalho.

Outras técnicas de inteligência artificial estão sendo estudadas, o que certamente irá contribuir com o surgimento de novos caminhos de pesquisa na área de segurança da informação. A utilização de redes neurais para a detecção de intrusos baseados em tráfego de rede, é uma das técnicas que devem ser testadas possibilitando a criação de um sistema de detecção de intrusos híbrido envolvendo os dois métodos de análise, baseado em rede e baseado em host, ambos utilizando redes neurais.

7.1. Direcionamentos futuros

A estrutura da rede pode ser modificada para apresentar uma melhor performance para o mesmo número de usuários e comandos.

A questão referente à quantidade de usuários e a performance da rede neural em identificar anomalias é muito importante, uma vez que para cada usuário que for acrescentado ao teste, teremos uma unidade de saída a mais, exigindo não somente um novo treinamento da rede, mas a alteração da estrutura desta.

Quando forem poucos usuários a serem acrescentados, é muito provável que não tenhamos muita alteração no desempenho da rede neural, mas seria bastante interessante verificar o desempenho deste modelo com centenas de usuários, como é o caso do laboratório utilizado neste trabalho, se fossem monitorados todos os alunos, totalizando quase 500 usuários em período de aulas.

A avaliação da rede também deve ser testada por período mais longo para descobrir com qual freqüência os usuários mudam seu comportamento, exigindo adaptações ao modelo de rede neural proposto para que esta rede possa se adaptar as pequenas alterações de comportamento dos usuários no dia-a-dia. A apresentação de uma arquitetura *back-propagation* pode auxiliar neste propósito.

Outro grande desafio, é montar uma estrutura de rede neural para detectar intrusão baseada no tráfego de rede, avaliando os protocolos contidos nos pacotes de dados que estão trafegando no segmento de rede, para descobrir ataques conhecidos e comparar com a detecção de IDS já existentes baseados em assinatura, como é o caso do Snort. Este tipo de detecção, baseado em rede, é bastante indicada para detectar ataques externos.

A união destes dois métodos de detecção de intrusão, baseado em comportamento e baseado no tráfego de rede, podem ser complementares e garantir mais confiabilidade as indicações dos sistemas de detecção de intrusão incrementando a segurança dos sistemas de rede.

8. REFERÊNCIAS BIBLIOGRÁFICAS

BACE, R.; *A New Look at Perpetrators of Computer Crime.* Em Anais do 16o. Department of Energy Computer Security Conference, 1994.

CANSIAN, Adriano Mauro; *Desenvolvimento de um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores*. Dissertação de Mestrado; Instituto de Física de São Carlos, USP; São Carlos; 1997.

CERT – Computer Emergency Response Team; Disponível em: <http://www.cert.org>

CHAPMAN, D. Brent; ZWICKY, Elizabeth D.; COOPER, Simon; *Construindo Firewalls Para a Internet;* 2a. Ed.; Ed. Campus; Rio de Janeiro; 2000.

CROSBIE, Mark; SPAFFORD Eugene; *Active defense of a computer system using autonomous agents;* Technical Report; COAST Group, department of Computer Sciences, Purdue University, West Lafayette, Indiana, 1995. Disponível em <`http://www.ceri as.purdue.edu/homes/spaf/tech-reps/9508.p`s>; Acesso em: 20 abr. 2002.

CAMPELLO, Rafael S.; WEBER, Raul F.; *Sistemas de Detecção de Intrusão*. Livro Texto dos Minicursos do 19 o Simpósio Brasileiro de Redes de Computadores, Florianópolis/SC, 2001.

DEBAR, H.; BECKER, M.; SIBONI, D.; *A Neural Network Component for an Intrusion Detection System*; Em Anais do IEEE Computer Society Symposium on Research in Security and Privacy; 1992.

DENNING, Dorothy E.; *An Intrusion-Detection Model*; Em IEEE Transactions on Software Engineering, Number 2, Vol. Se-13. 1987.

GARFINKEL, Simson; SPAFFORD Eugene; *Pratical Unix and Internet Security.* O'Reilly and Assossiates, Sebastopol, California, 1996.

GIL, Antônio de Loureiro; *Segurança em Informática;* Ed. Atlas; São Paulo; 1994.

GNU General Public License. Disponível em <http://www.gnu.com/copyleft/gpl.txt>; Acesso em: 12 jan. 2002.

HAYKIN, S.; *Neural Networks: A Comprehensive Foundation.* MacMillan College Publishing Co., New York, Ny, USA, 1994.

HEADY, R.; LUGER, G.; MACCABE A.; SERVILLA M.; *The Architecture of a Network Level Intrusion Detection System, Technical Report;* Departament of Computer Science; University of New Mexico; USA; 1990

HOWARD, John D.; LONGSTAFF, Thomas A.; *A Common Language for Computer Security Inicdents;* Sandia National Laboratories; 1998

Internet Consortium Security Agency; Disponível em: <http://www.icsa.net>; Acesso em: 12 jan.2002.

JACOBSON, Van; LERES, Craig; McCANE, Steven; Lawrence; Berkeley National Laboratory, 1994; Disponível em <http://www-nrg.ee.lbl.gov>; Acesso em: 10 abr.2002.

JAIN, A. K.; MAO, J.; MOHIUDIUMM, K. M.; *Artificial Neural Networks: A Tutorial.*; IEEE Computer. 1996.

KO, C.; FRINCKE, D. A.; GOAN Jr, T.; HEBERLEIN, L. T.;LEVITT, K.; MUKHERJEE, B.;WEE, C.; *Analysis if an Algorithm for Distibuted Recognation and Accountability.* Em Anais do First ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 1993.

LAMPSON, B.W.; *Protection Operating Systems*; Review. 1974.

LUNT, T. F.; *Detecting Intruders in Computer Systems;* Em Conference on Auditing and Computer Technology; 1993.

McCULLOCH, W. S.; PITTS, W.; *A Logical Calculus of Ideas Immanet in Nervous Activity.* Bulletim of Mathematical Biophysics. Vol.5. 1943.

MEDEIROS, Carlos D. R.; *Segurança da Informação - Implantação de Medidas e Ferramentas de Segurança da Informação;* Trabalho de Conclusão de Curso; Univille Universidade da Região de Joinville; Joinville, SC; 2001

NEUMANN P.; PARKER, D.; *A Summary of Computer Misuse Techniques.* Em Anais of 12th National Computer Security Conference, 1989.

PORRAS, P.A.; NEWMANN, P.G. *EMERALD: Event monitoring enabling response to anomalous live disturbances.* Em: National Information Systems Security Conference (NISSC), 20., Baltimore. Proceedings... S.n.:S.l.], 1997.

POW, Keesje Duarte; *Segurança na Arquitetura TCP/IP. De Firewalls a canais Seguros;* Dissertação de Mestrado; Universidade Estadual de Campinas; Campinas, SP; 1999;

RANUM, Marcus J.; *Firewalls FAQ - Frequently Asked Questions.* 1995.

RUSSEL, D; GANGEMI Sr., G. T.;*Computer Security Basics.* O'Reilly and Associates, Sebastopol, California, 1991.

RYAN, Jake; LIN Meng-Jang; MIIKKULAINEN, Risto; *Intrusion Detection with Neural Networks;* Em Advances in Neural Information Processing Systems 10, Cambridge, MA; 1998.

SANTOS, Ricardo B. dos; CAMINHAS, Valmir M.; ERRICO, Luciano de; *Detecção de Intrusos: Uma Abordagem Usando Redes Neurais*; Universidade Federal de Minas Gerais; Disponível em <http://www.ufmg.br> Acesso em mai. 2002

SOARES, Luiz F. G.; LEMOS, Guido; COLCHER, Sérgio; *Redes de Computadores: Das LANs, MANs e WANs às Redes ATM*; 2 ª Edição; Ed. Campus, Rio de Janeiro, 1995.

STANIFORD-CHEN, S.; TUNG, B.; SCHNACKENBERG, D.; *The Common Intrusion Detection Framework;* Information Survivability Workshop, Orlando, FL, USA; 1998

TAN, Kymie; *The Application of Neural Networks to Unix Computer Security;* Technical Report; Computer Science Department; University of Melbourne; Austrália; 1995.

TANENBAUM, Andrew S.; *Redes de computadores;* 3ª. edição, Ed. Campus; Rio de Janeiro; 1994.

VENEMA, W.Z.; *TCP Wrapper: Networking Monitoring, Access Control and Booby Traps;* Em Anais of Third UNIX Security Symposium, Baltimore, USA, 1992

WEST-BROWN, Moira J.; STIKVOORT, Don; KOSSAKOWSKI, Klaus-Peter; *Handbook* for *Computer Security Incident Response Teams;* Carnegie Mellon University, Software Engineering Institute; 1998; Disponível em http://sei.cmu.edu/pub/documents/98.reports/pdf/98hb001.pdf; Acessado em jan. de 2002;